REAL TIME SIMULATION AND CONTROL 3000 TON SURFACE EFFECT SHIP

Thomas Spencer Nelson

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

REAL TIME SIMULATION AND CONTROL
3000 TON SURFACE EFFECT SHIP

by

Thomas Spencer Nelson, III

December 1979

Thesis Advisor: A. Gerba, Jr.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Real Time Simulation and Control of 3000 Ton Surface Effect Ship		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Thomas Spencer Nelson, III		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1979
		13. NUMBER OF PAGES 134
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Surface Effect Ship
Real Time Simulator
Hybrid

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A real time, man controlled simulator exhibiting the characteristics of a 3000 Ton Surface Effect Ship (3K-SES) was developed using a set of simplified 5 degree of freedom equations of motion.

A software program was developed to compute and present to the SES pilot on a real time graphics output

(20. ABSTRACT Continued)

"alpha-numeric-perspective" visual cues pertaining to the vehicle states.

Man generated control responses during 3K-SES accidental effector nozzle failure tests and the requirement for computer assist control systems during the recovery phase of such failures was examined.

Real Time Simulation and Control
3000 Ton Surface Effect Ship

by

Thomas Spencer Nelson, III
Major, United States Marine Corps
B.S., North Carolina State University, 1966

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1979

ABSTRACT

A real time, man controlled simulator exhibiting the characteristics of a 3000 Ton Surface Effect Ship (3K-SES) was developed using a set of simplified 5 degree of freedom equations of motion.

A software program was developed to compute and present to the SES pilot on a real time graphics output "alpha-numeric-perspective" visual cues pertaining to the vehicle states.

Man generated control responses during 3K-SES accidental effector nozzle failure tests and the requirement for computer assist control systems during the recovery phase of such failures was examined.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A_{ws}	Average sidewall wetted area, starboard side	ft^2
A_{wp}	Average sidewall wetted area, port side	ft^2
A_{31}	Added mass coefficient in roll force equation	ft-slug
A_{33}	Added mass coefficient in pitch force equation	ft-slug
A_{w2}	Average wetted sidewall area of the bow	ft^2
A_{w1}	Average wetted sidewall area of the stern	ft^2
A_{22}	Added mass coefficient in yaw force equation	slug s^2 lb_f/ ft
β	Sideslip angle	rad
C_{DX}	Coefficient drag on x-direction	$lb_f s^2 lb_m/ft^2$
C_{DY}	Coefficient drag in y-direction	$lb_f s^2 lb_m/ft^2$
C_{DZP}	Sidewall roll moment lumped parameter coefficient	$lb_f s^2/ft^2$
C_{DP}	Bow pitch force lumped parameter coefficient	non-dimensional
F_{sw}	Sidewall roll force	lbs
F_{ss}	Sidewall starboard buoyancy force	lbs
F_{sp}	Sidewall port buoyancy force	lbs
F_1	Stern buoyancy force	lbs
F_2	Bow seal pitching force	lbs
F_3	Bow buoyancy force	lbs
g	Gravitational acceleration	ft/s^2
I_X	Moment of inertia about x-axis	slug ft^2
I_Y	Moment of inertia about y-axis	slug ft^2

I_z	Moment of inertia about z-axis	slug ft^2
J_1	Track deviation performance index	ft^2 sec
J_2	Control effort performance index	rad^2 sec
J_3	Overall performance index	non-dimensional
K	Summation of moments about x-axis	lbs-ft
ℓ_{sw}	Length of sidewall	ft
ℓ_{dp}	Actual draft of port sidewall	ft
ℓ_w	x-direction displacement of hull drag centroid	ft
ℓ_{d_1}	Average draft of SES sidewall	ft
ℓ_d	Average draft of bow seal	ft
ℓ_x	Length from center of gravity to stern	ft
ℓ_{x_1}	Average draft of sidewall	
ℓ_{31}	Pitch moment lever arm for bow sidewall buoyancy force	ft
ℓ_3	Pitch moment lever arm for bow seal force	ft
m	Mass of the rigid ship	ft
M	Summation of moments about y-axis	lbs-ft
N	Summation of moments about z-axis	lbs-ft
P_b	Plenum pressure	lbs ft^2
p	Lumped drag centroid point	non-dimensional
p'	Lumped drag centroid point	non-dimensional
$\dot{p}$	Roll acceleration	rad/sec^2
q	Pitch rate	rad/sec
$\dot{q}$	Pitch acceleration	rad/sec^2
r	Yaw rate	rad/sec
$\dot{r}$	Yaw acceleration	rad/sec^2

s_1	Turning moment lever arm of no. 1 engine	ft
s_2	Turning moment lever arm of no. 2 engine	ft
s_3	Turning moment lever arm of no. 3 engine	ft
s_4	Turning moment lever arm of no. 4 engine	ft
T_7	Total thrust magnitude on no. 1 engine	lbs
T_8	Total thrust magnitude on no. 2 engine	lbs
T_9	Total thrust magnitude on no. 3 engine	lbs
T_{10}	Total thrust mangitude on no. 4 engine	lbs
T_{forw}	Total forward thrust vector of effectors	lbs
T_{side}	Total side thrust vector of effectors	lbs
T_{yaw}	Total turning moment generated by effectors	lbs ft
u	Velocity in x-direction (surge)	ft/sec
$\dot{u}$	Acceleration in x-direction	ft/sec^2
v	Velocity in y-direction (sway)	ft/sec
$\dot{v}$	Acceleration in y-direction	ft/sec^2
V_s	Total velocity	ft/sec
w_e	Width of bow seal	ft
X	Summation of forces in x-direction	lbs
X_o	X_{NAV} coordinate of SES	ft
$\dot{X}_o$	X_{NAV} velocity of SES	ft
Y	Summation of forces in y-direction	lbs
Y_o	Y_{NAV} coordinates of SES	ft
$\dot{Y}_o$	Y_{NAV} velocity of SES	ft
δ	Effector angle commanded	rad
δ_7	Effector angle of no. 1 nozzle	rad
δ_8	Effector angle of no. 2 nozzle	rad

δ_9	Effector angle of no. 3 nozzle	rad
δ_{10}	Effector angle of no. 4 nozzle	rad
ψ	Heading angle of SES	rad
ϕ	Roll angle of SES	rad
θ	Pitch angle of SES	rad
ρ	Density of sea water	lbm/ft^3

I. INTRODUCTION

Extensive analysis of the dynamics of the Captured Air Bubble (CAB) Surface Effect Ship (SES) has been accomplished in the past. Such analysis has produced equations of motion of the SES type vehicle in six degrees of freedom [Ref. 1]. Analysis at the Naval Postgraduate School has centered around the XR-3 and 100-B SES [Refs. 2,3], with current analytical effort being directed toward the 3K-SES [Ref. 4]. The primary tool for study of the 3K-SES characteristics is the highly comprehensive five degree of freedom (DOF) SES DBS Program developed by the David W. Taylor Naval Ship Research and Development Center [Ref. 5]. A version of this program used at NPS is called DBSIM5D. This data base program allows very detailed study to be conducted on the 3K-SES response to precomputed control inputs such as step or ramp thruster commands and their associated accidental failure possibilities.

A natural extension in the study of control of the 3K-SES is to evaluate the vehicle's response to an on-going "man generated" control input on a real time basis. The rationale for developing a "man-in-the-loop" SES simulator would be to 1) test the man in responding to accidental failure situations that may be experienced during operational conditions, 2) train personnel in operator control techniques unique to SES type vehicles, 3) provide a means

to test hardware to be utilized in SES craft, and 4) provide a design tool for development of control schemes or system structural changes to improve performance.

Accordingly a simplified, non-linear five degree of freedom Real Time Simulator (RTS5D) for the 3K-SES was developed such that a continuous, on-going, observable, real-time solution is interfaced with man-generated control inputs.

II. EQUATIONS OF MOTION

A. COORDINATE SYSTEMS AND ASSUMPTIONS

It was recognized that the RTS5D would require accurate and yet simplified equations of motion when compared to those of the more extensive DBSIM5D. The rationale for requiring simplification was to start with a right hand side (RHS) that could be utilized effectively in a real time simulation. To satisfy this requirement a model was selected that would use point source of drag forces consisting of only a few lumped-coefficient terms. The equations used are based upon those developed in Ref. 1 which follow this concept. Additional assumptions used in the simplified 5 DOF equations were as follows:

1) All accelerations are measured at the center of gravity.
2) Vehicle is "free-to-heave".
3) All cross coupled moments of inertia are zero.
4) All cross products of angular velocities in force equations are zero.
5) All roll and pitch angles used in force equations are subject to small angle approximations.
6) Calm water conditions.
7) Mass and mass distribution of vehicle are constant.
8) Effect of changes of aerodynamic force are neglected.

Utilizing the above assumptions, the simplified 5 DOF equations of motion are developed using the coordinate systems shown in Fig. 1 and Fig. 2, the force vector diagrams shown in Fig. 3, Fig. 4A, Fig. 4B, and the following equations from Newton's laws of motion:

SURGE	$m(\dot{u} - vr)$	=	X
SWAY	$m(\dot{v} - ur)$	=	Y
YAW	$I_Z \dot{r}$	=	N
PITCH	$I_Y \dot{q}$	=	M
ROLL	$I_X \dot{p}$	=	K

where:

m	=	mass of the rigid ship	, slugs
u	=	velocity in x-direction (SURGE)	, ft/sec
v	=	velocity in y-direction (SWAY)	, ft/sec
r	=	angular velocity about the z-axis	, deg/sec
p	=	angular velocity about the x-axis	, deg/sec
q	=	angular velocity about the y-axis	, deg/sec
I_Z	=	moment of inertia about the z-axis	, lb_m-ft^2
I_Y	=	moment of inertia about the y-axis	, lb_m-ft^2
I_X	=	moment of inertia about the x-axis	, lb_m-ft^2
X	=	summation of forces in x-direction	, lb_f
Y	=	summation of forces in y-direction	, lb_f
N	=	summation of moments about the z-axis	, ft-lb_f

M = summation of moments about the y-axis, $ft\text{-}lb_f$

K = summation of moments about the x-axis, $ft\text{-}lb_f$

Additionally the following navigation relationships are utilized (see Fig. 1).

$$\dot{X}_o = u \cos \psi - v \sin \psi$$

$$\dot{Y}_o = u \sin \psi + v \cos \psi$$

$$V_s = u^2 + v^2$$

$$\beta = \tan^{-1} (-v/u)$$

where:

ψ = heading angle

β = drift angle

δ = thrust vector angle

Also,

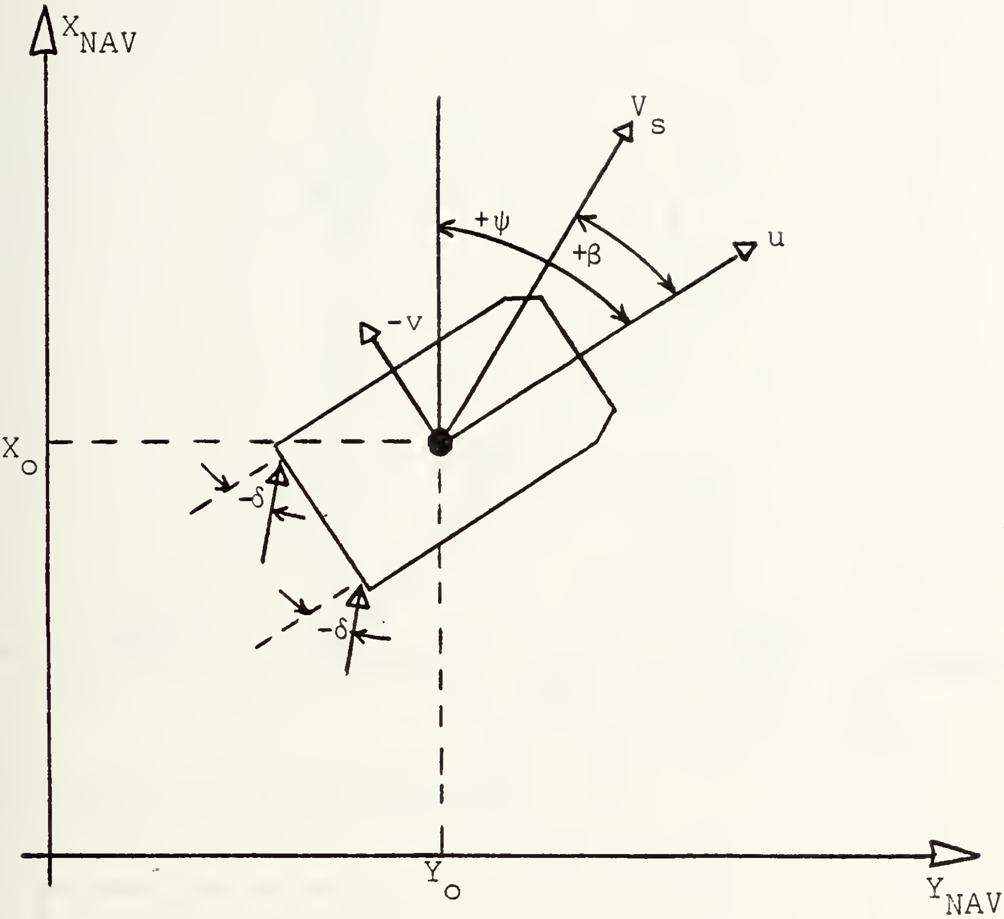
$$X_o = \int \dot{X}_o \, dt \qquad u = \int \dot{u} \, dt$$

$$Y_o = \int \dot{Y}_o \, dt \qquad v = \int \dot{v} \, dt$$

$$r = \int \dot{r} \, dt \qquad \psi = \int r \, dt$$

$$p = \int \dot{p} \, dt \qquad \phi = \int p \, dt$$

$$q = \int \dot{q} \, dt \qquad \theta = \int q \, dt$$

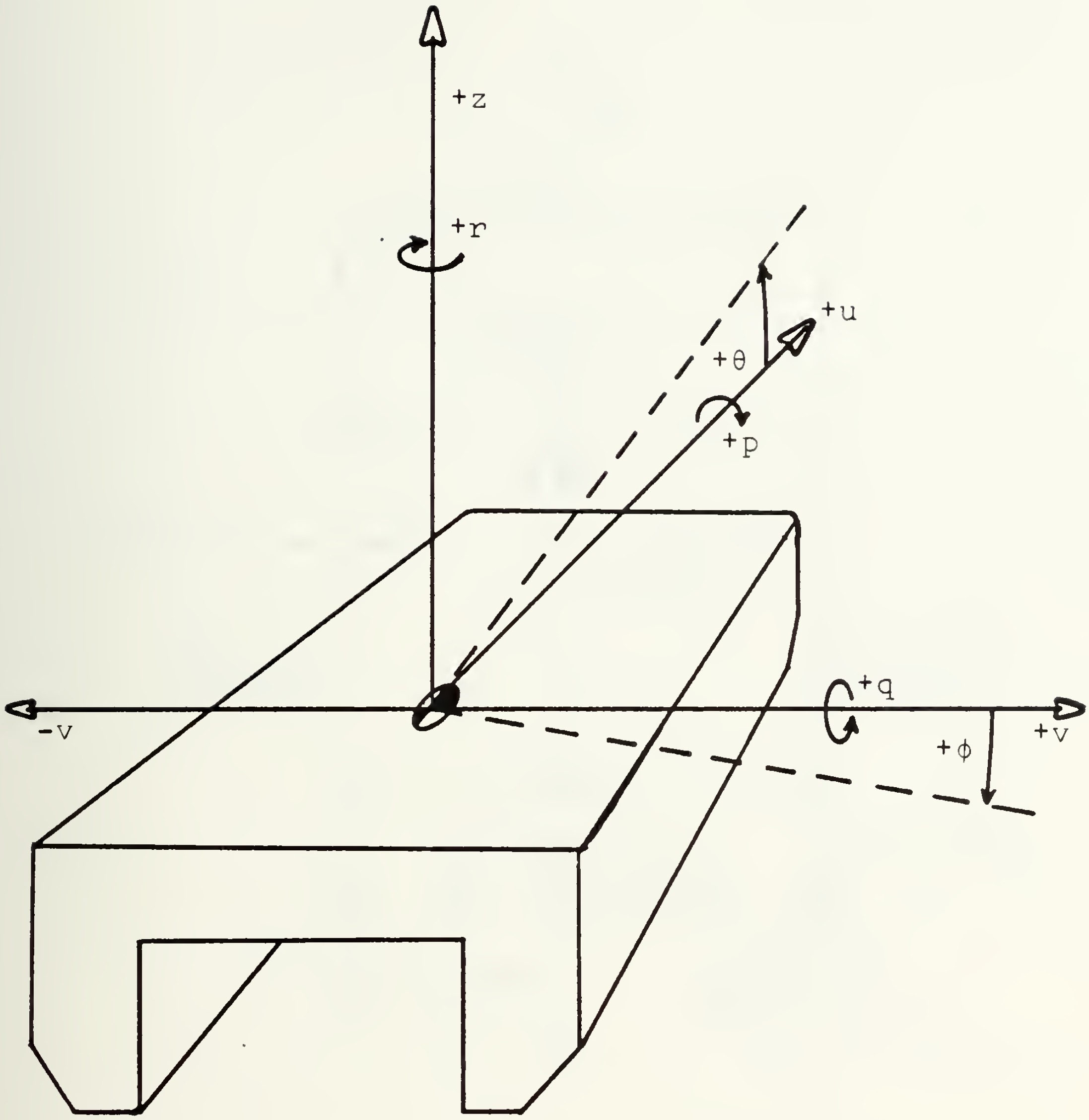


NOTE: 1) $\psi=0$ when vehicle heads parallel to + X_{NAV} axis.

2) Vehicle in figure above shown with negative thrust vector angle δ, positive ψ, negative β, negative sway velocity v, i.e., in a right turn.

Figure 1

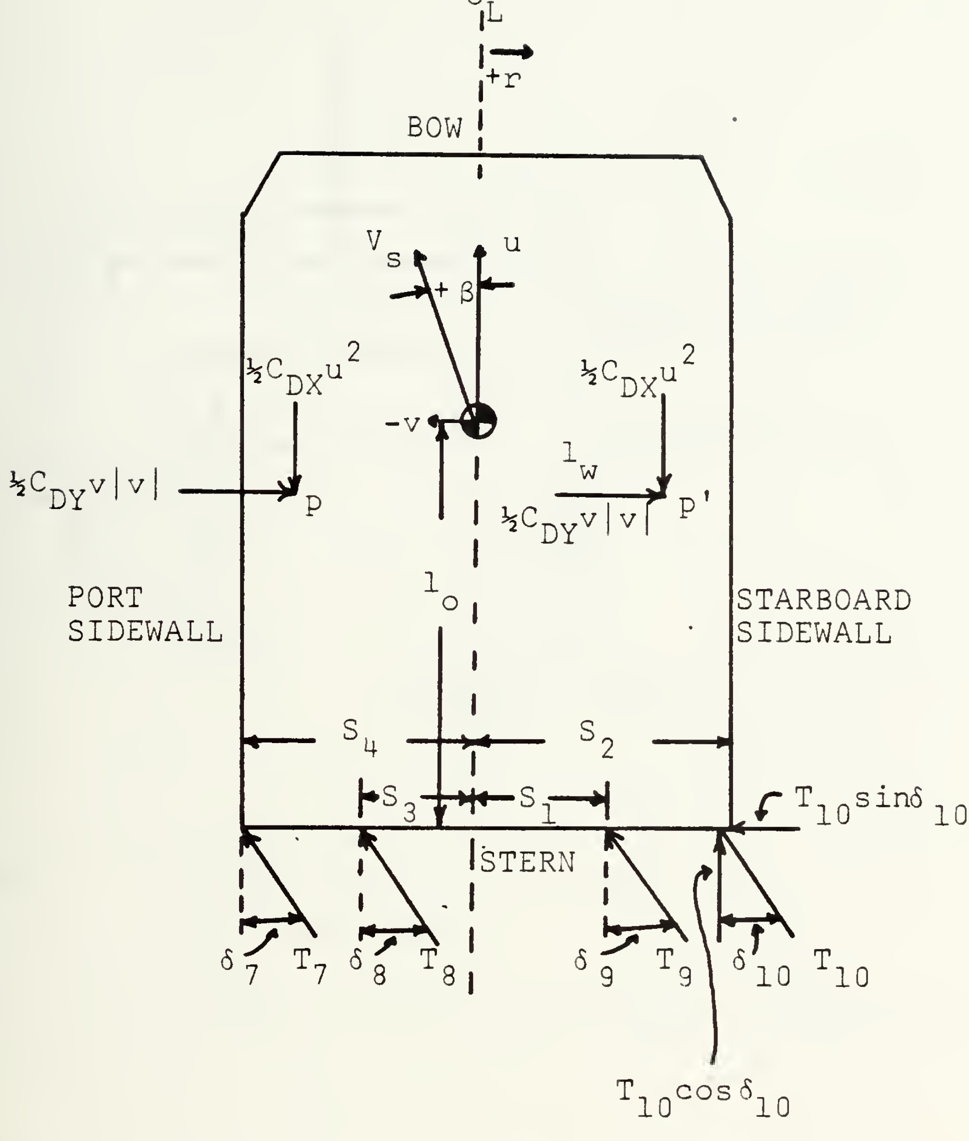
Definition Of Coordinate System (Part I)



NOTE: Direction for positive velocity and angular rates are shown

Figure 2

Definition Of Coordinate System (Park II)



NOTE: 1) p and p' are equivalent point force centroids.

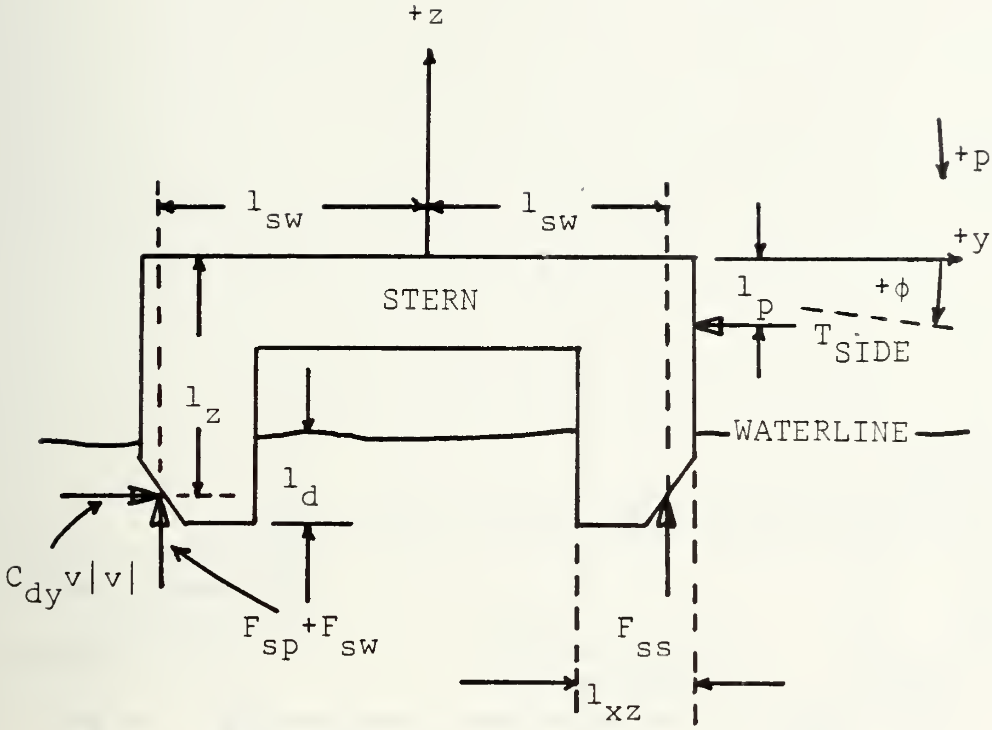
2) δ_7, δ_8, δ_9, δ_{10} are thrust vector angles and are not required to be equal.

3) T_7, T_8, T_9, T_{10} are thrust magnitudes and are not required to be equal.

4) Force directions shown are for a right turn

Figure 3

Surface Effect Ship (Top View)

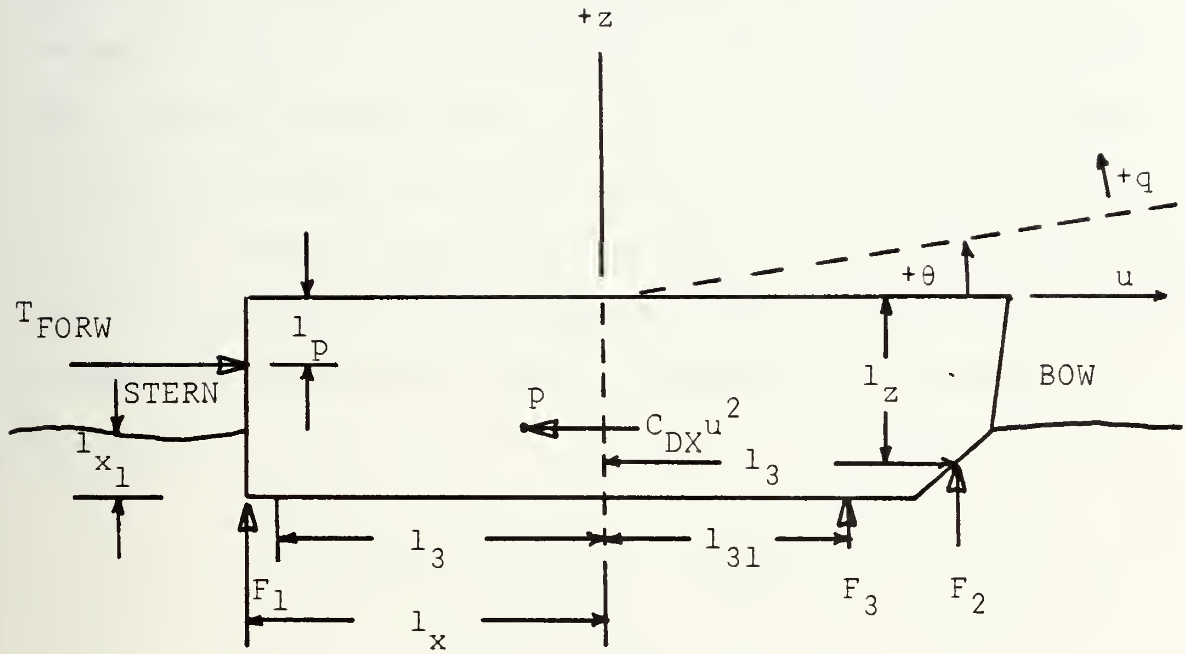


NOTE: 1) Right Turn Forces Acting, i.e., δ_7, δ_8, δ_9, δ_{10} are negative

2) $T_{SIDE} = T_7 \sin\delta_y + T_8 \sin\delta_8 + T_9 \sin\delta_9 + T_{10} \sin\delta_{10}$

Figure 4A

Surface Effect Ship (Stern View)



NOTE: 1) $T_{FORW} = T_7 \cos \delta_7 + T_8 \cos \delta_8 + T_9 \cos \delta_9 + T_{10} \cos \delta_{10}$

(see Fig. 3)

Figure 4B

Surface Effect Ship (Side View)

B. FORCES AND MOMENTS

1. Surge Forces (See Fig. 4B)

The forward acceleration was determined to be generated by the summation of thrust vectored in the forward direction T_{forw}. The resultant acceleration of the SES was assumed to be opposed by a retarding force exhibiting a "velocity-squared" characteristic. Additionally, Newton's laws of motion specified a "v·r" product that contributed to the drag component of the equation when the vehicle was in a turn (centrifugal force reaction term). The simplified equation of motion for the surge acceleration follows from a summation of forces in the u direction as depicted in Fig. 4B.

$$\dot{u} = (T_{forw}/m) - (C_{DX}u^2/m) + vr$$

2. Sway Forces (See Fig. 4A)

The side acceleration of the vehicle was analyzed to be the result of the summation of the thrust vectored parallel to the vehicle's y-axis and a retardation "velocity squared" phenomena such as that produced by the cross flow drag term of the sidewall as described in Ref. 6. This retardation force is augmented by a "u·r" product as specified by Newton's Law.

$$\dot{v} = T_{side}/m - C_{DY}\, v|v|/m - ur$$

Note the requirement to force the $C_{DY}v^2/m$ term to maintain the sign of the sway velocity such that both left and right turns may be computed.

3. Yaw Moments (See Fig. 3)

The yaw acceleration was a turning moment summation which was defined to be influenced by three primary forces operating over moment arms. It is of interest to note that the third term is an "added mass" term which is considered to operate on the same pressure point as the $C_{DY}\, v|v|$ term. The term T_{yaw} is a summation of moments generated by the four thrusters as shown in Fig. 3.

$$\begin{aligned} T_{yaw} = {} & s_4 T_7 \cos \delta_7 + s_3 T_8 \cos \delta_8 - s_1 T_9 \cos \delta_9 \\ & - s_2 T_{10} \cos \delta_{10} - \ell_o T_7 \sin \delta_7 - \ell_o T_8 \sin \delta_8 \\ & - \ell_o T_9 \sin \delta_9 - \ell_o T_{10} \sin \delta_{10} \end{aligned}$$

$$\dot{r} = C_{DY}\, \ell_w\, v|v|/I_Z + T_{yaw}/I_Z + A_{22} uv\ell_w/I_Z$$

4. Pitch Moments (See Fig. 4B)

The pitch acceleration was assumed to be the summation of moments generated by forward thrust, T_{forw}, the buoyancy of the sidewalls of the SES, F_1 and F_2, and a vertical force generated at the bow of the vehicle, F_3. This vertical bow force was defined to be a lumped parameter

term which modeled the reaction force due to plenum pressure acting against the bow seal.

ℓ_d = average draft of bow seal

$A_{w_1} = \ell_x \ell_{x_1} + ((\ell_x \tan\theta)/2)\ell_x$

$A_{w_2} = \ell_x \ell_{x_1} - ((\ell_x \tan\theta)/2)\ell_x$

$F_1 = A_{w_1} \ell_{x_2} \rho g$

$F_2 = A_{w_2} \ell x_2 \rho g$

$F_3 = C_{dp}\bar{p}_b w_e (\ell_d - \ell_{31} \tan\theta)$

ℓ_{x_1} = average draft of sidewall

p_b = plenum pressure

A_{w_1} = average wetted sidewall area of the stern

ℓx_2 = width of one sidewall

w_e = width of bow seal

A_{w_2} = average wetted sidewall area of the bow

A_{33} = added mass coefficient

ρ = water density

$g = 32.3\ ft/s^2$

ℓ_{31} = lever arm of bow seal

ℓ_3 = lever arm of buoyancy force

$$\dot{q} = (T_{forw}\,\ell_p + F_3\ell_{31} + F_2\ell_3 - C_{DX}\,u^2\ell_z$$

$$- F_1\ell_3 - A_{33}uq)/I_Y$$

Note the added mass term $A_{33}uq$ which was required to provide damping to the pitch moment and is specified in Ref. 1.

It was found that the response of the SES to a pitch perturbation without the added mass component was undamped and approximately sinusoidal. Unlike the yaw equation where the added mass term was a small contributor to damping, the pitch added mass term was found to be of significant importance in modeling the known pitch motion.

5. Roll Moments (See Fig. 4A)

The roll acceleration equation was analyzed to be a summation of moments generated by buoyancy forces of the port and starboard sidewalls, F_{sp} and F_{ss}, the thrust vectored parallel to the y-axis ofthe vehicle, T_{side}, a side force $C_{DY}\,v|v|$, and a lumped coefficient vertical force, F_{sw}. The vertical force F_{sw} was defined as the force generated in a turn due to the sidewall curvature (dead rise angle) acting against the cross flow of water.

$$F_{sp} = \rho g\ A_{wp} \ell_{dp}$$

$$F_{ss} = \rho g\ A_{ws}\ \ell_{ds}$$

$$F_{sw} = C_{DZP}\ v|v|$$

ℓ_{d_1} = average draft of SES sidewall

$$\ell_{dp} = \ell_{d_1} - \ell_{sw} \tan\phi$$

$$\ell_{ds} = \ell_{d_1} + \ell_{sw} \tan\phi$$

A_{wp} = average wetted area port

A_{ws} = average wetted area starboard

$$\dot{p} = ((F_{sp} - F_{ss})\ell_{sw} - T_{side}\ \ell_p + C_{DY}\ v|v|\ \ell_z$$

$$- F_{sw}\ \ell_{sw} - A_{31}\ up)/I_X$$

Again note the added mass term $A_{33}up/I_X$ which was found to be essential in modeling the damping phenomena of the roll motion.

In summary, the simplified 5 DOF equations of motion used in RTS5D for the 3K TON SES are:

<u>SURGE</u> $\dot{u} = T_{forw}/m - C_{DX}u^2/m + vr$ [Fig. 4B]

<u>SWAY</u> $\dot{v} = T_{side}/m - C_{DY}\ v|v|/m - ur$ [Fig. 4A]

$$\underline{\text{YAW}} \qquad \dot{r} = T_{yaw}/I_Z + C_{DY}\, v|v|\, \ell_w/I_Z + A_{22} uv\ell_w/I_Z \qquad \text{[Fig. 3]}$$

$$\underline{\text{PITCH}} \qquad \dot{q} = (T_{forw}\, \ell_p + F_3 \ell_{31} + F_2 \ell_3 - C_{DX}\, u^2 \ell_z$$

$$- F_1 \ell_3 - A_{34} uq)/I_Y \qquad \text{[Fig. 4B]}$$

$$\underline{\text{ROLL}} \qquad \dot{p} = ((F_{sp} - F_{ss})\ell_{sw} - T_{side}\, \ell_p + C_{DY}\, v|v|\ell_z$$

$$- F_{sw}\ell_{sw} - A_{33}\, up)/I_X \qquad \text{[Fig. 4A]}$$

D. PARAMETER IDENTIFICATION

These equations are an extension of the 3 DOF flat turn SES model developed by Gerba and Thaler in Ref. 4. The identification of craft parameters C_{DX}, C_{DY}, and $_w$ is described in Ref. 4 and repeated here for completeness.

The surge drag coefficient C_{DX} is determined by selecting a steady state turn condition, where T_{forw}, u, v, r, and m are known.

$$\dot{u} = 0 = \frac{T_{forw}}{m} - \frac{C_{DX} u^2}{m} + vr$$

from which

$$C_{DX} = \frac{T_{forw} + mvr}{u^2}$$

The sway drag coefficient C_{DY} is determined by the same steady state turn condition which requires

$$\dot{v} = 0 = \frac{T_{side}}{m} - \frac{C_{DY}}{m} v|v| - ur$$

where

$$C_{DY} = \frac{T_{side} - mur}{v|v|}$$

The sway drag moment arm follows utilizing the yaw acceleration equation

$$\dot{r} = 0 = \frac{T_{yaw}}{I_Z} + \frac{C_{DY} \; v|v| \ell_w}{I_Z} + \frac{A_{22} \; uv \; \ell_w}{I_Z}$$

which yields

$$\dot{w} = \frac{-T_{yaw}}{C_{DY} \; v|v| + A_{22}uv}$$

The addition of the pitch and roll equations introduces additional lumped coefficients C_{DP} and C_{DZP}. These are easily solved using known constants ℓ_w, ℓ_p, C_{DY}, ℓ_z, ℓ_{sw}, w_e, p_b, ℓ_d and steady state values of F_{sp}, F_{ss}, F_1, F_2, T_{forw}, T_{side}, u, v, and r. It is significant to note that in a steady state condition the angular roll rate, p, and angular pitch rate, q, are both required to equal zero.

This is in contrast to the yaw rate, r, where a finite steady state value is desired in a turn. Thus the added mass terms of $A_{33}uq$ and $A_{31}up$ in the pitch and roll equations are not utilized in the determination of craft parameters; their function is strictly confined to damping of their respective accelerations. Therefore, utilizing the pitch equation with known steady state turn values,

$$\dot{q} = 0 = (T_{forw}\,\ell_p + C_{DP}p_b w_e(\ell_d - \ell_{31}\tan\theta)\,\ell_{31}$$

$$+ F_2\ell_3 - C_{DX}u^2\ell_2 - F_1\ell_3)/I_Y$$

yields

$$C_{DP} = \frac{C_{DX}u^2\ell_z + F_1\ell_3 - F_2\ell_3 - T_{forw}\,\ell_p}{p_b w_e(\ell_d - \ell_{31}\tan\theta)\ell_{31}}$$

The roll equation under steady state turn conditions yields

$$\dot{p} = 0 = ((F_{sp} - F_{ss})\ell_w - T_{side}\,\ell_p + C_{DY}\,v|v|\,\ell_z$$

$$- C_{DZP}\,v|v|\,\ell_{sw})/I_Z$$

from which

$$C_{DZP} = ((F_{sp} - F_{ss})\ell_w - T_{side}\,\ell_p + C_{DY}\,v_{ss}|v_{ss}|\ell_z)/$$

$$(v_{ss}|v_{ss}|\,\ell_{sw})$$

where

F_{sp} = port sidewall buoyancy in pounds [Fig. 4A]

F_{ss} = starboard sidewall buoyancy in pounds [Fig. 4A]

T_{side} = thrust vectored parallel to vehicles y-axis in pounds [Fig. 3]

v_{ss} = sway steady state velocity in feet/sec

C_{DY} = sway drag coefficient for steady state condition

ℓ_w = sway drag moment arm in feet [Fig. 3]

ℓ_p = effector thrust moment arm in feet [Fig. 4A]

ℓ_z = surge drag moment arm in feet [Fig. 4A]

$_{sw}$ = sidewall buoyancy force moment arm in feet [Fig. 4A]

III. INTEGRATION METHOD

A. ERROR ANALYSIS

A rectangular (RECT) type integration was used to compute the velocity and positional variables of the force equations. Selection of the type of integration over a more complex type such as Runga Kutta, Variable Step enabled the loop delay time due to integration to be held to a minimum.

An analysis of the error introduced into the solution of the simplified 5 DOF equations through the use of rectangular integration vice Runga-Kutta, Variable Step was made. The simplified 5 DOF equations of motion were programmed on the NPS IBM 360/67 computer using the Continuous System Modeling Program (CSMP). One series of computer runs was made utilizing rectangular integration over a spectrum of step sizes ranging from 0.01 secs to 0.25 secs. A second computer run was made utilizing Runga-Kutta, Variable Step type integration. The steady state results are shown in Table I. Examination of the final value magnitudes established that no significant steady state error was due to rectangular integration for the time increments utilized.

Transient error of the variables, r, ϕ, and θ was examined. A series of computer runs using incrementally

TABLE I. STEADY STATE ERROR ANALYSIS FOR RECTANGULAR INTEGRATION

Type Integration	DELT	U	V	R	PHI	THETA	X_o	Y_o
Runga Kutta Variable Step	VAR	81.08	6.9447	-.03880	-.01798	.02035	2058	-915
RECT	.050	80.77	6.9446	-.03895	-.01778	.02035	2055	-931
RECT	.075	80.75	6.9447	-.03896	-.01775	.02035	2046	-927
RECT	.1	80.78	6.9446	-.03895	-.01780	.02035	2044	-933
RECT	.125	80.75	6.9450	-.03897	-.01746	.02034	2072	-954
RECT	.15	80.73	6.9448	-.03898	-.01729	.02034	2063	-965
RECT	.20	80.69	6.9449	-.03900	-.01703	.02037	2078	-990
RECT	.25	80.75	6.9447	-.03897	-.01731	.02034	2055	-958

Note: 1) Step effector angle input of 25°

2) Velocity (u) of 56 knts.

larger rectangular integration step sizes was compared against an identical run using Runga-Kutta Variable Step type integration. The first peak absolute magnitude results are shown in Table II. The RTS5D had an iteration time of .084 seconds therefore the "transient" magnitudes were in error by approximately 2% to 3% due to rectangular integration.

TABLE II. TRANSIENT FIRST PEAK ERROR ANALYSIS FOR RECTANGULAR INTEGRATION

DELT	r	ϕ	θ
.050	1.42%	1.64%	2.58%
.075	1.89%	2.41%	2.84%
.100	2.16%	3.38%	3.25%
.125	2.35%	3.65%	8.57%

Note:

1) Step effector angle input of 25° and velocity (u) of 56 knts.
2) All percentages are referenced to Runga-Kutta variable step magnitudes.

IV. IMPLEMENTATION

A. PHILOSOPHY

Given the decision to develop a real time motion analysis simulator for the 3K-SES, the various options available were examined. Prime requirements for such a simulator were the necessity for real clock interfacing, suitable input-output ports, and a dedicated Central Processing Unit (CPU).

The "all analog" computer simulation was particularly attractive from the aspect of interfacing man generated control inputs into an on-going real time solution. With the analog approach instantaneous solutions are obtained from patchboard amplifiers. Two considerations which did not lend themselves well to the all analog technique were how to maintain high accuracy solutions in wide ranging variables when constrained to a voltage limit of plus or minus 100 volts, and how to output the system variables in such a manner that a man could physically interpret what was happening and respond with the required corrective, timely control effort. Additional supplementary problems associated with the "quarter-square" multiplication error and amplifier drift/run duration limitations gave rise to examining other possible approaches.

The "all digital" computer simulation offered high accuracy with wide ranging variables and a capability to

interface with two graphics display consoles. Additionally a line printer was available to provide "hard copy" of variables of interest as the solution was being processed. Two major disadvantages were the serial and discrete nature of the integration method of a digital computer, and the lack of high voltage, high accuracy potentiometer controls to allow the man to influence the motion of the vehicle as the solution was being presented to him.

A desire to take advantage of the best features of both the analog and digital alternatives led to a decision to develop the real time simulator (RTS5D) of the 3K-SES utilizing the simplified 5 DOF equations of motion developed in Section II.

B. REQUIREMENTS

The following criteria were required of the RTS5D:

1) The simplified equations of motion would be solved and the results output on a real time basis.
2) The solution would be subject to real time control efforts generated by a man observing the output.

Development of the simplified 5 DOF equations of motion and the use of rectangular type integration for simulating the 3K-SES ensured that loop delay attributable to digital computational needs would be minimal and thus the solution would be compatible to a real time environment.

The requirement to output the results on a real time basis such that a man observed and responded with real time

control efforts necessitated the use of graphics terminals interfaced with a digital computer. Large volumes of data and graphical cues were presented to the pilot and updated on a real time basis in much the same manner as found on the bridge consoles of the 3K-SES shown in Ref. 7. Software algorithms designed to present "alpha-numeric" data and graphic perspectives corresponding to the vehicles current state variables enabled the observer to respond on a real time basis.

Real time man generated control requirements necessitated the need for a hybrid environment and its associated interface between a digital computer and an analog computer. The RTS5D utilizes four (4) computer systems: One (1) medium size general purpose digital computer (XDS-9300), two (2) small graphics digital computers (AGT-10), and one (1) analog computer (CI5000). The XDS-9300 has a CPU capable of handling 32K words of main (core) memory with a magnetic drum for secondary storage. The XDS-9300 may be programmed in assembly language or FORTRAN. The CI-5000 is an analog computer with a full complement of digital logic. Hardwired trunk lines linking the XDS-9300 and CI-5000 allow hybrid computer operations. The two AGT-10's are small general purpose digital computers with 8K of main memory and magnetic disk for secondary storage. The AGT-10 systems have the capability of accepting input and of displaying output on their CRT screens. Both the CI-5000

and the AGT-10 have the capability of communicating with the XDS-9300 directly, but cannot communicate with each other except via the XDS-9300 [Ref. 8].

C. HARDWARE DESCRIPTION

Hardware layout for the RTS5D is shown in Figure 5A. A "joystick" type high voltage potentiometer attached to the pilot's seat was hardwired to the CI-5000. This joystick served the purpose of the pilot's effector angle controller. Left or right stick transmitted an appropriate nozzle deflection angle in the form of a voltage level to the CI-5000 where it was quantized in an analog to digital process and introduced into the XDS-9300's ongoing solution of the simplified 5 DOF equations. In like manner the voltage levels of six other "dial" potentiometers used to control thruster output levels and program modes were introduced into the solution process as deemed appropriate by the operator. The six potentiometers were located on a "thruster console" (See Fig. 5B) within reach of the pilot (See Fig. 5A). Additional hardware included a standard digital computer line printer, an analog 8 channel strip chart recorder, and an analog X-Y plotter.

D. SOFTWARE DESCRIPTION

1. Introduction

The RTS5D software package was a FORTRAN IV digital computer program which consisted of a main program and two

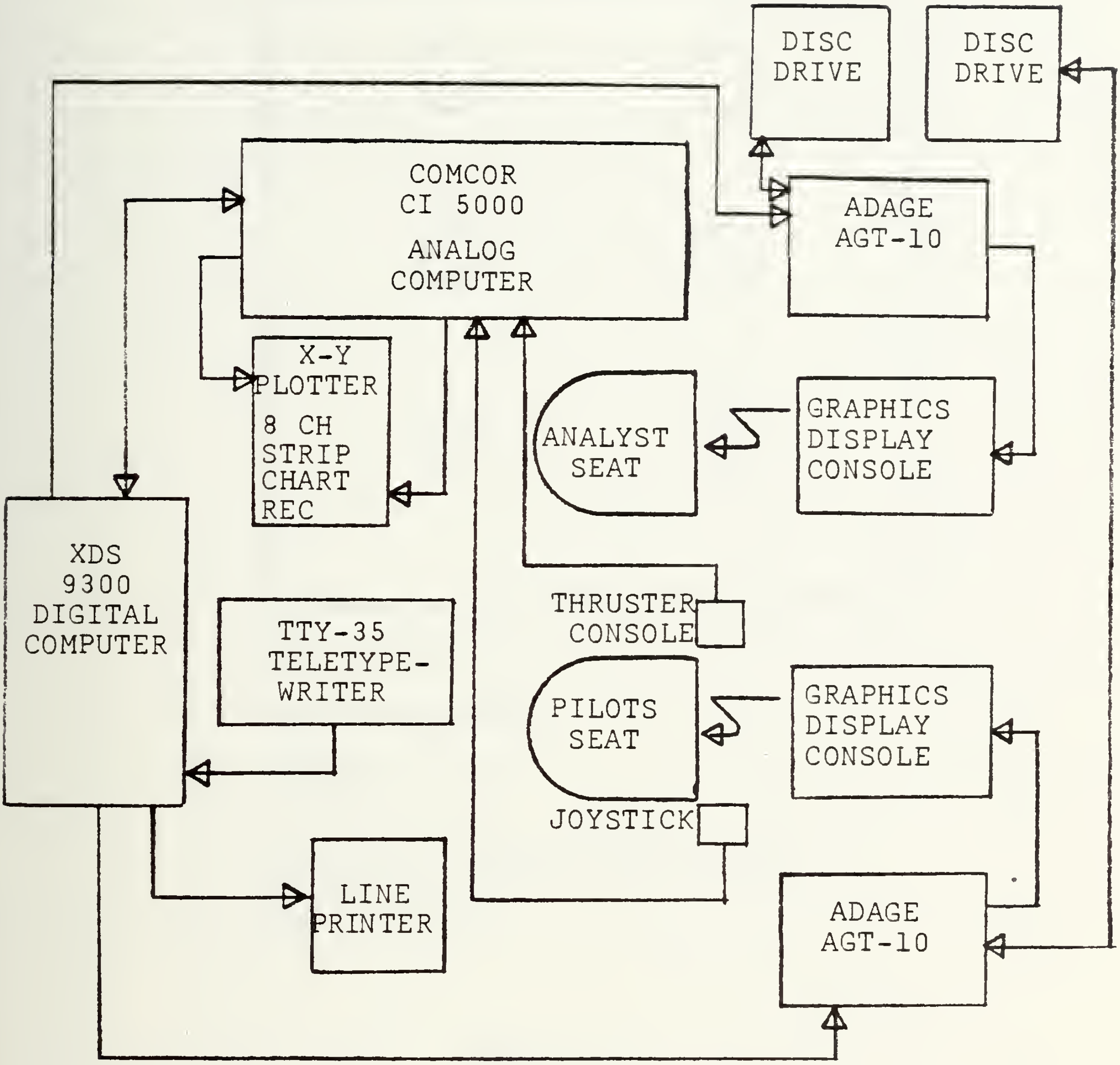
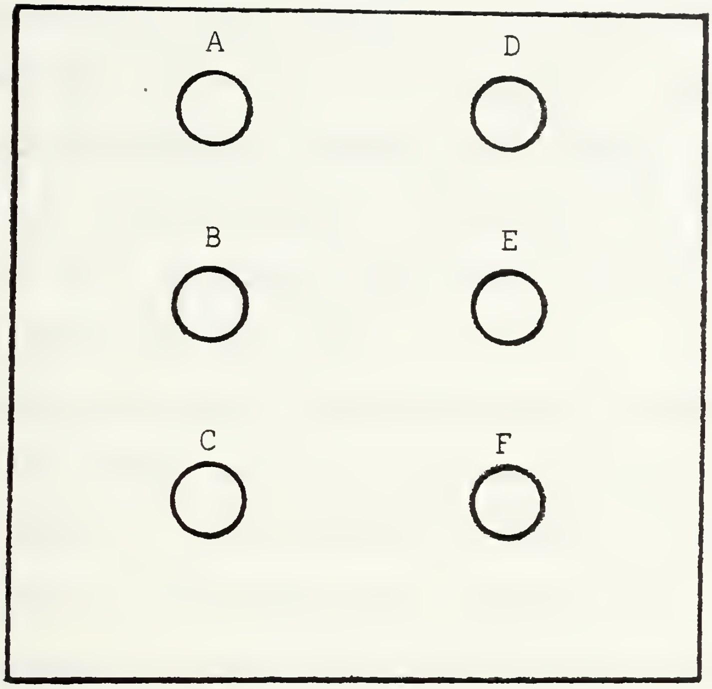


Figure 5A
RTS5D Block Diagram



A - Thruster T-7

B - Run, Hold, Restart

C - Thruster T-8

D - Thruster T-9

E - Accidental Failure Nozzle Angle Controller For Thruster T-10

F - Thruster T-10

Figure 5B

Thruster Console

major subroutines. The program gave the operator the option to start, stop, and hold the dynamics solution process. The software package controlled a multiplexing algorithm used on the graphics display consoles and sampled all analog control and timing inputs. A program flow chart is shown in Fig. 6A and a detailed flow chart for the graphics multiplexing is shown in Fig. 6B.

The program began by initializing all program variables and parameters and then requested from the operator any parameter changes. The request and input of changes were made on a TTY-35 Teletypewriter (See Fig. 5A). The program then started the real time 1000 Hz clock timing circuits in the CI-5000 analog computer and entered the main program iteration loop. In the main loop the sampling of analog control inputs by the pilot was accomplished by subroutine ADDA. Craft force equations were computed and integrated. A perspective view of the simplified form of the SES and an electronic road were utilized in the RTS5D to give real time visual cues to the pilot on the status of the vehicle's states. This section of the display gave the pilot his primary guidance information and was updated on the pilot's graphics display console on every iteration of the main program loop. The program then evaluated whether an accidental failure mode had been selected by the operator. If the failure mode was selected the program searched the craft's states to see if the recovery criteria

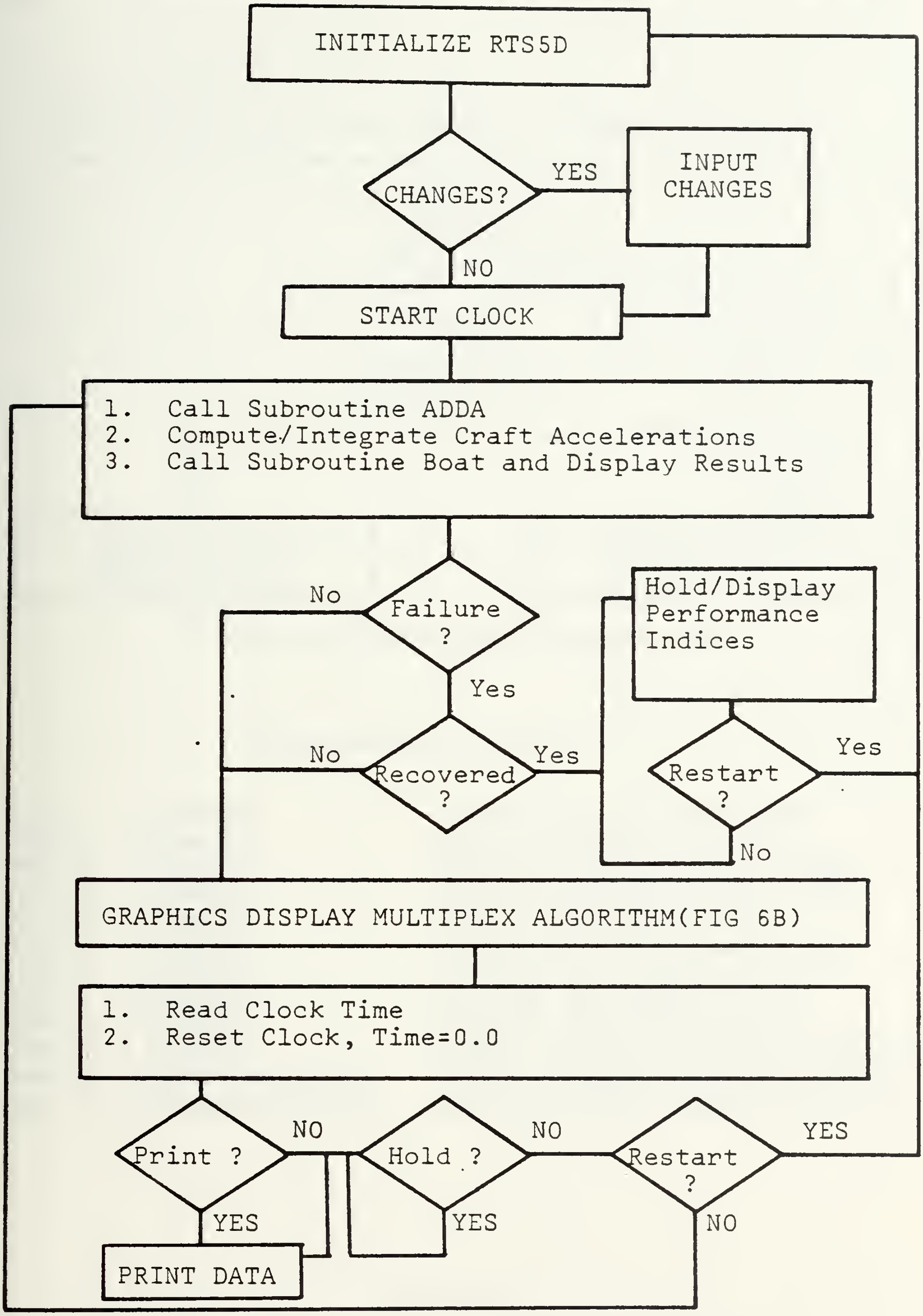
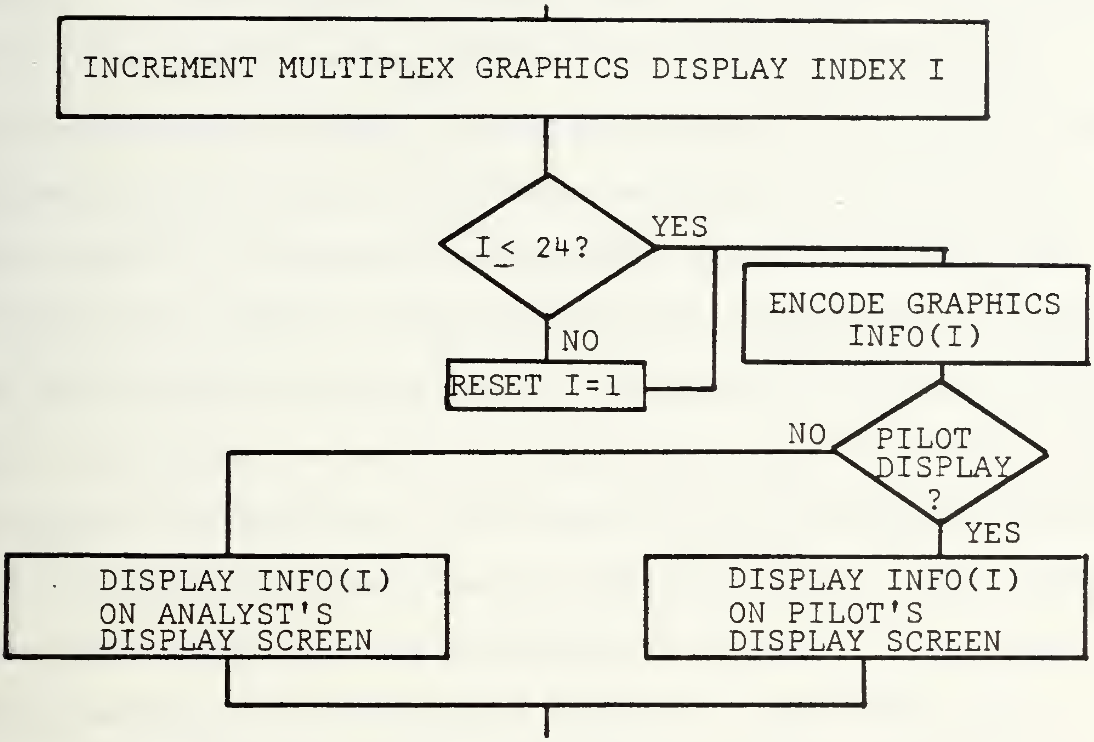


Figure 6A

RTS5D Program Flow Chart



DESCRIPTION OF INFO(I)

INFO(1) u DYNAMICS
INFO(2) $\dot{u}$ DYNAMICS
INFO(3) v DYNAMICS
INFO(4) $\dot{v}$ DYNAMICS
INFO(5) r DYNAMICS
INFO(6) $\dot{r}$ DYNAMICS
INFO(7) p DYNAMICS
INFO(8) $\dot{p}$ DYNAMICS
INFO(9) q DYNAMICS
INFO(10) $\dot{q}$ DYNAMICS
INFO(11) POSITION MAX VALUES
INFO(12) POSITION MIN VALUES
INFO(13) VELOCITY MAX VALUES
INFO(14) VELOCITY MIN VALUES
INFO(15) ACCELERATION MAX VALUES
INFO(16) ACCELERATION MIN VALUES
INFO(17) NAVIGATION DATA
INFO(18) EFFECTOR/THRUSTER DATA
INFO(19) SEA TRACK NAV PLOT
INFO(20) SEA TRACK NAV PLOT
INFO(21) PERFORMANCE INDICES DATA
INFO(22) PRESENT POSITIONAL VARIABLES
INFO(23) PRESENT VELOCITY VARIABLES
INFO(24) PRESENT ACCELERATION VAR.

Figure 6B

RTS5D Graphics Display Multiplex Algorithm

had been satisfied through the maneuvering inputs of the pilot. If the recovery criteria was fulfilled, the program went into a hold status and awaited further instructions by the operator on when to restart the run. If the recovery criteria had not been fulfilled the program continued the main loop by incrementing a graphics multiplex index and entering the graphics display multiplex algorithm to search for and display the appropriate information array associated with the current value of the graphics index I (See Fig. 6B). A graphics multiplexing algorithm was established to reduce the iteration time required for the main loop. Each information array contained approximately 50 elements associated with positions on the display screens. Each array had a display time which was required to encode and project the array on the screen. This display time varied directly with the number of words in the array. The prioritization of the graphic display console information arrays was made such that the perspective of the SES and the electronic road were displayed on each iteration and were not subject to the program's graphics display multiplexing routine. All other information arrays were multiplexed on a time share basis such that on any main loop only two information arrays were projected on the screen, the SES/road and the array designated by the graphics display index I. This concept was mandatory due to the large volume of data/ graphics being presented. It is instructional to recall

the fact that all variables were computed and updated in the memory of the XDS-9300 on <u>every</u> loop, whereas the display routine was multiplexed on a time share basis. The main loop iteration time was 84 msec. Information arrays in the multiplex routine were updated on the graphics display consoles every 2.0 sec. On exit from the graphics display multiplex algorithm the program read the value of the real time clock and then reset the clock to zero. From this the Δ time magnitude was determined. Following this the program searches for print, restart, and hold flags which are set by the operator. The program loops back to the AD/DA subroutine and begins a new iteration.

2. <u>Graphics Techniques</u>

The two AGT-10 Graphics Display Consoles were utilized to display real time data to the operator such that he could control the craft. The process of transmitting and displaying desired data to an AGT-10 was accomplished utilizing FORTRAN callable graphics subroutines in the XDS-9300. One graphics subroutine TEXTO was used to transmit alpha-numeric data and one graphics subroutine GRAPHO was used to transmit graphical data.

Subroutine TEXTO transmitted alpha-numeric data to the AGT-10 screen. The positioning of the desired data on the screen itself was accomplished by utilizing a TEXTO screen coordinate system of dividing the 12 inch by 12 inch AGT-10 screen into 40 equally spaced lines from top to bottom with each line having room for 96 individual

characters on it as shown in Fig. 7A. Subroutine TEXTO statements analogous to FORTRAN "WRITE" and "FORMAT" statements were used to specify the letter character or variable magnitude to be displayed and their position on the display screen.

Subroutine GRAPHO was used to transmit the "picture" portion of the output to the display screen. The coordinate system utilized by GRAPHO is a normalized X-Y system with the ordinate and abscissa intersecting in the center of the screen as shown in Fig. 7B. A point on the GRAPHO coordinate system corresponding to (1.,0.) would be located at the right edge of the screen centered between the top and bottom of the screen. A line could be generated on the screen by specifying the line start coordinates, line stop coordinates and commanding a "draw". The AGT-10 would connect the two designated points and a line would be displayed on the screen. Concurrently if only two point sources of light had been desired to be displayed, the command "jump" could be utilized such that the AGT-10 would not connect the second coordinate point with the first coordinate point. In this manner a series of commands was generated via subroutine GRAPHO whereby a picture was drawn on the AGT-10 screen using an array of sequential coordinate points and their respective "draw" or "jump" instruction.

3. Analyst Graphic Display Console

The display presented to the analyst during the conduct of a run is shown in Fig. 8. The magnitudes of the

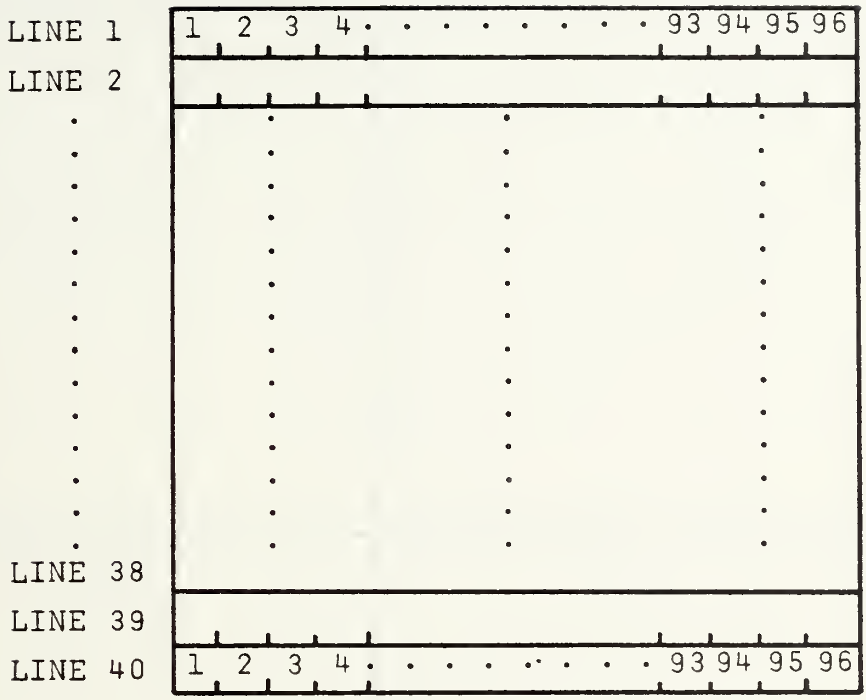


Figure 7A
Subroutine Texto Coordinate System

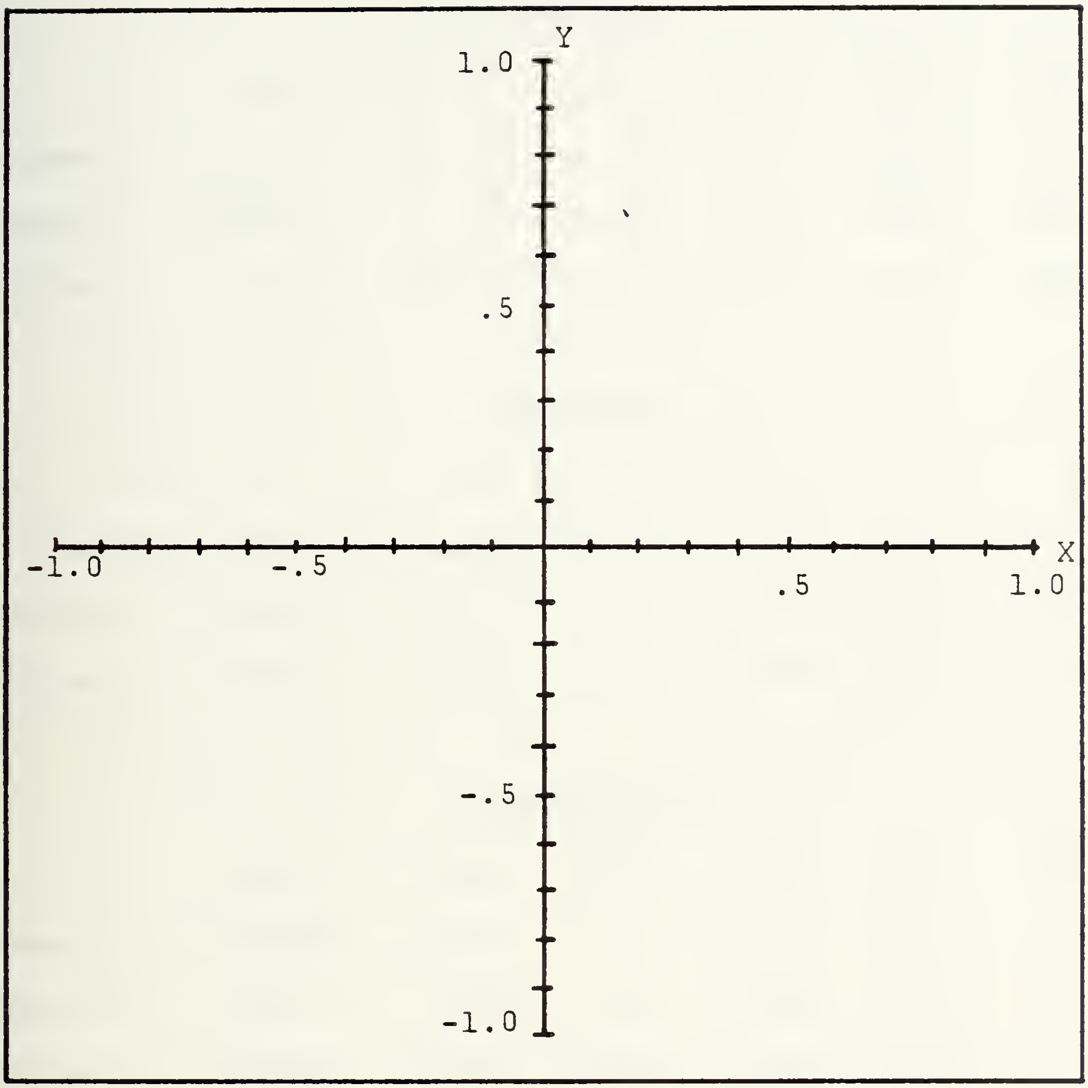


Figure 7B

Subroutine Grapho Coordinate System

VARIABLE STATUS

POSITION

	X	Y	ZZ	PSI	PHI	THETA
Present	2193.1	-2172.0	0.0	-0.6	-.014	.005
Maximum	2193.1	0.0	0.0	0.0	.009	.007
Minimum	0.0	-2172.0	0.0	-0.6	-.016	.000

VELOCITY

	U	V	W	R	P	Q
Present	87.6	4.86	0.0	-.01	.002	+0.00
Maximum	94.1	5.18	0.0	0.0	+.161	+.110
Minimum	87.6	0.0	0.0	-.012	-.305	-.034

ACCELERATION

	UDOT	VDOT	WDOT	RDOT	PDOT	QDOT
Present	-0.29	-0.05	0.0	+0.00	.001	+.00
Maximum	0.0	0.833	0.0	0.09	.003	+.00
Minimum	-0.36	-0.06	0.0	-0.01	-.004	-.00

Note: Typical run values shown

Figure 8

Analyst Graphic Display Console

the variables were computed by the XDS-9300 on every iteration of the program. The XDS-9300 then followed a display multiplexing algorithm (See Fig. 6B) which updated the variables every 2. seconds. The variables were categorized by position, velocity, and acceleration. Each category had three lines of data displayed. The "present" data line displayed the real time magnitude of the variable column it was located in. The "maximum" and "minimum" were the max and min magnitudes that had been computed for the variable during that particular run. At the start of a new run all variables are initialized (See Fig. 6A).

4. <u>Pilot Graphics Display Console</u>

The display presented to the pilot during the conduct of a run is shown in Fig. 9A and Fig. 9B. The pilot display contained three major sections, VARIABLE DYNAMICS, SEA TRACK/ EMERGENCY PERFORMANCE INDEX, CRAFT PERSPECTIVE/NORTH ROAD/ NAV DATA.

a. Variable Dynamics

The time histories of the accelerations and their respective velocities were presented in graphical form. An acceleration plot and its respective velocity plot shared the same electronic strip chart such that one was superimposed over the other. The acceleration plots were displayed as "light" toned lines whereas the velocity plots were displayed as "heavy" toned lines. As the RTS5D program incremented in real time, it presented an incremental

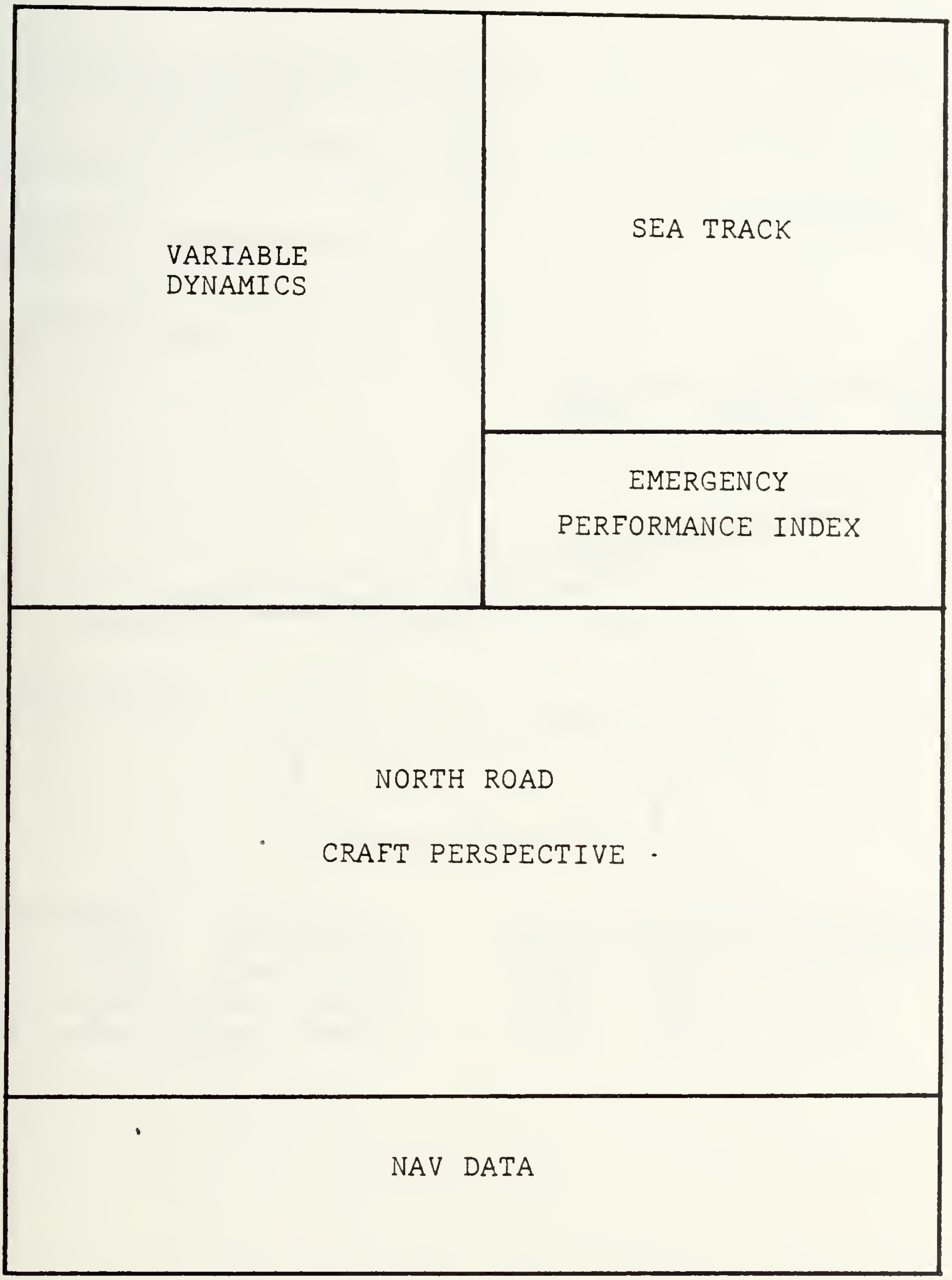


Figure 9A
Pilot Graphics Display Sections

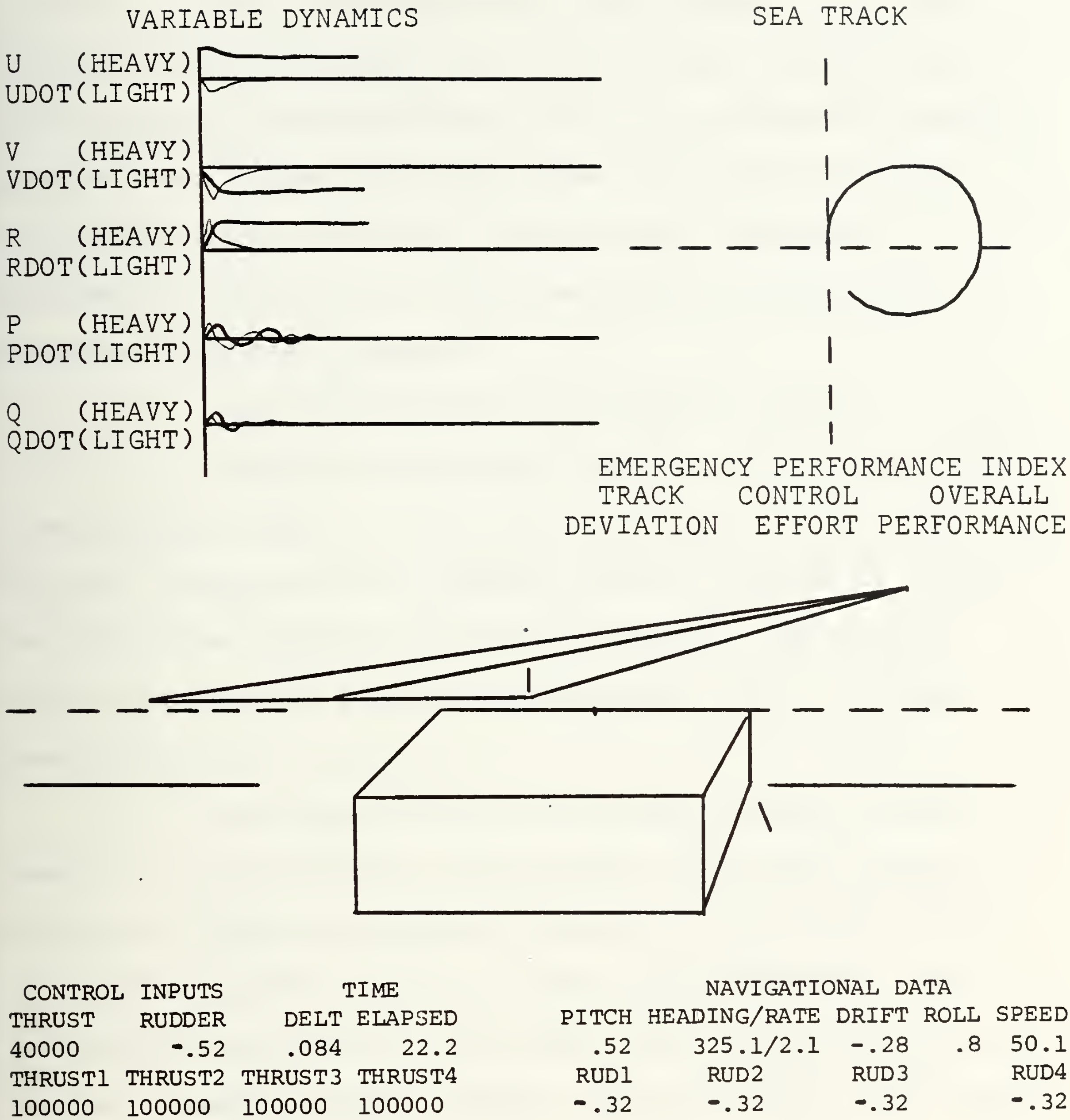


Figure 9-B
Pilot Graphic Display

increase in each displayed state time history. The time histories would progress in a left-to-right movement and upon reaching the right most limit of the VARIABLE DYNAMICS section were erased from the screen and restarted at the left side of the electronic strip chart. Execution of a maneuver of the vehicle would immediately be displayed in the time histories displays.

b. Sea Track/Emergency Performance Index

The sea track portion of this display area was a navigational plot of the ship trajectory (X_o, Y_o) (See Fig. 9B). This track was updated every 2 seconds as implemented in the XDS-9300 multiplex algorithm. The track would erase itself and begin anew at the craft's current position every 3.5 minutes.

The performance of the pilot during an accidental failure situation was measured and graded through the use of three performance indices J_1 (TRACK DEVIATION), J_2 (CONTROL EFFORT), and J_3 (OVERALL PERFORMANCE). All performance tests were initialized such that the vehicle was on a heading of 000° (NORTH) and on the X_{NAV} axis with a steady state velocity.

J_1, TRACK DEVIATION, was an "integral-error-squared-plus-time" performance index which penalized the distance that the pilot allowed the vehicle's c.g. to drift away from the X_{NAV} axis in addition to the length of time the vehicle remained off the X_{NAV} axis (See Fig. 10).

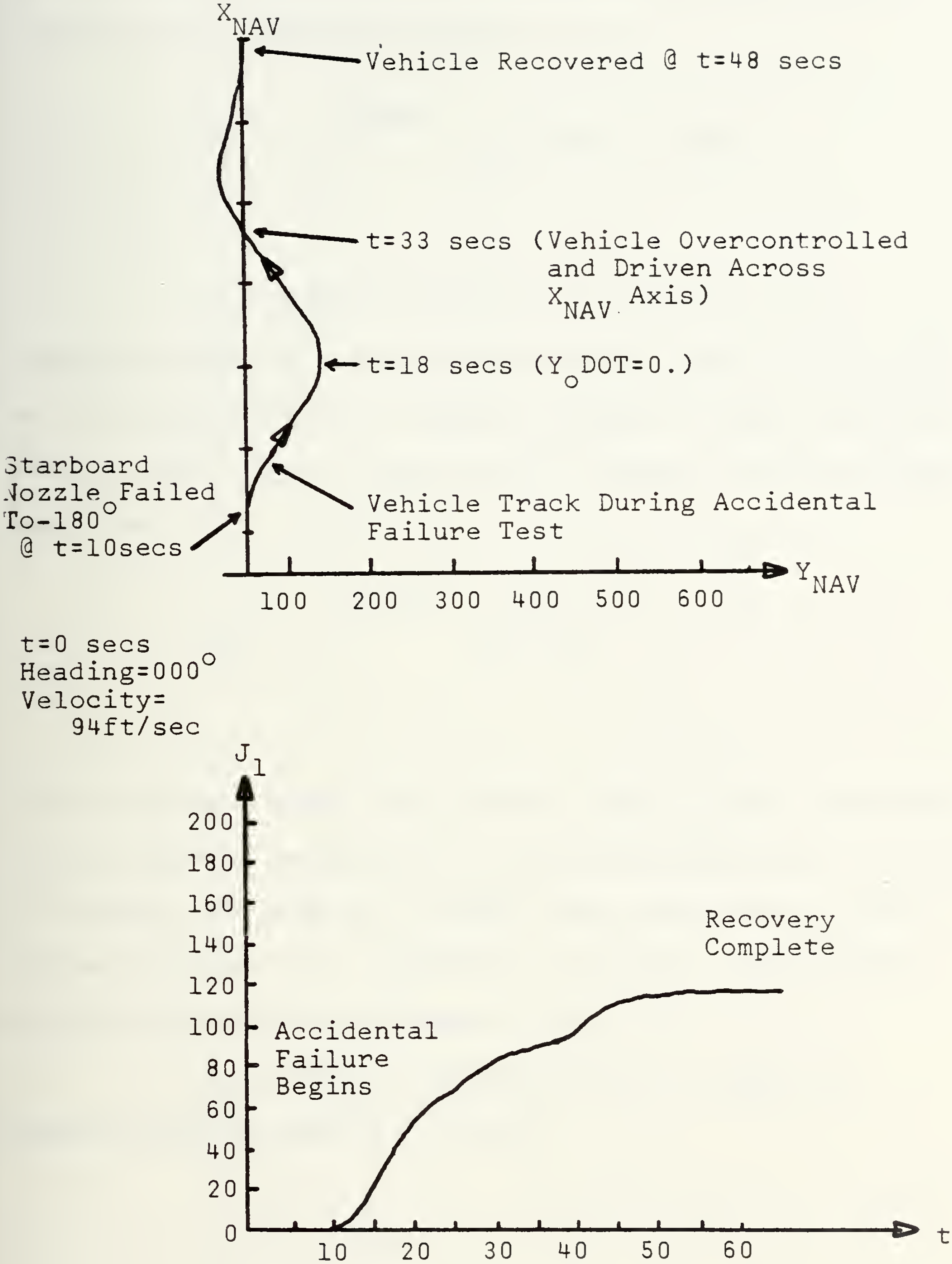


Figure 10

Track Deviation Performance Index(J_1)

At the systems entry into an accidental failure mode of operation the XDS-9300 computed J_1 as

$$J_1 = \int_{t_o}^{t_{final}} (Y_o)^2 \, dt + \text{TIME}$$

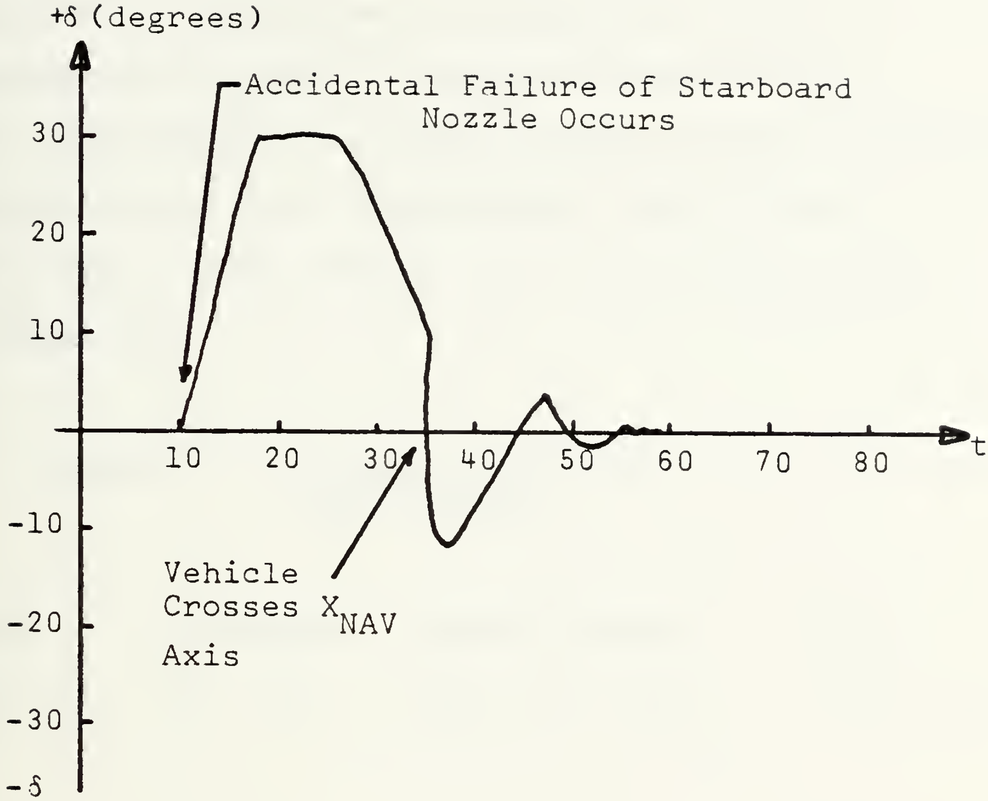
J_2, CONTROL EFFORT, was an index used to penalize the pilot for excessive control effort during the man controlled recovery maneuver segment of the accidental failure test. The J_2 index was an integral "error-squared" index, as

$$J_2 = \int_{t_o}^{t_{final}} (\delta)^2 \, dt$$

Note that the δ under the integral sign is the δ associated with the nozzle deflection of the three operational thrusters of the SES, and does not include the nozzle angle of the failed thruster (i.e., starboard thruster). A graphical interpretation of J_2 is shown in Fig. 11.

J_3, OVERALL PERFORMANCE, was a magnitude representing the sum of J_1 and J_2

$$J_3 = J_1 + J_2$$



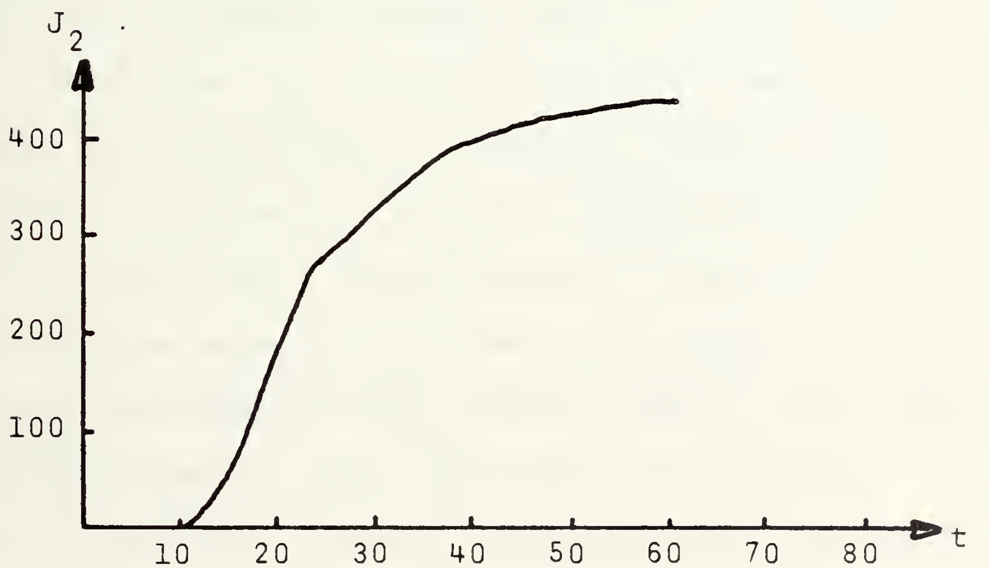


Figure 11
Control Effort Performance Index(J_2)

At the completion of the test the performance algorithm in the XDS-9300 was programmed to display the numeric scores representing J_1 and J_2 and the word evaluation, EXCELLENT, ABOVE-AVERAGE, or GOOD below the overall performance heading shown in Fig. 9B. The magnitude range of the performance index J_3 for each of the 3 word evaluations are shown in Table III.

TABLE III. OVERALL PERFORMANCE INDEX RATINGS

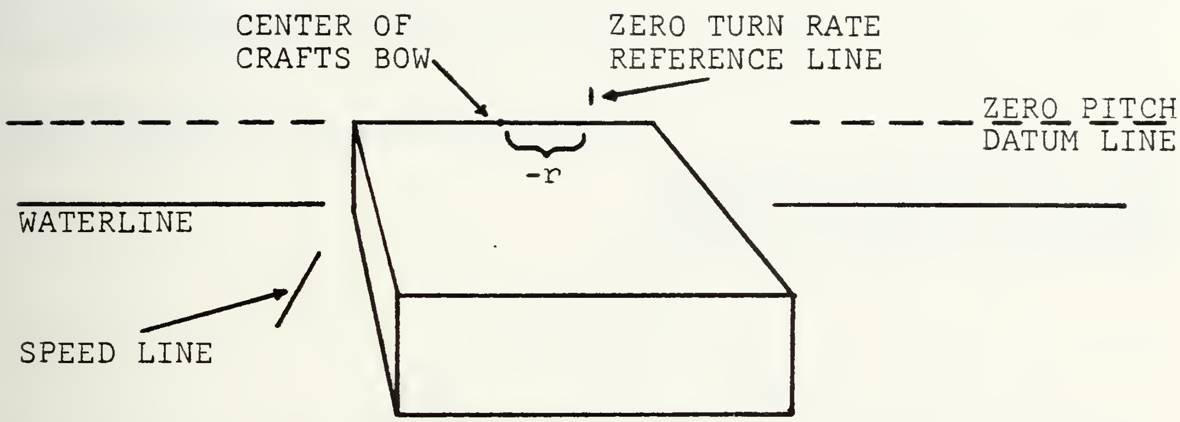
RATING	EXCELLENT	ABOVE AVERAGE	GOOD
RANGE	$J_3 < 250$	$250 \leq J_3 < 500$	$500 \leq J_3$

c. Craft Perspective/North Road/NAV Data

Real time visual cues to the pilot were generated on the lower half of the pilot's graphics display console in the form of a perspective view of a simplified shape of the craft from the rear looking down and forward onto the craft and the area immediately ahead of the craft's path. An electronic north-heading road shown forward of the craft provided information to assist the operator in proper control of the vehicle. Numeric navigation data concerning the craft's position and the status of the nozzle angles and effector thrust settings were presented in the area immediately below the craft perspective (See Fig. 9A and Fig. 9B).

The craft perspective view presented real time graphical information on the vehicle's turning rate (r), velocity (u), pitch angle (θ), and roll angle (ϕ). The electronic north road's centerline was the positive X_{NAV} axis (See Fig. 1). The north road was displayed on the pilot's screen whenever the pilot had maneuvered the craft to within 1500 feet of the X_{NAV} axis and the heading of the vehicle was on a northerly course. At other times the north road was not visible to the pilot. When displayed, the north road provided the pilot with visual cues on the vehicle's heading and navigation position with respect to the fixed X_{NAV} axis. When the pilot maneuvered the craft such that the north road was not visible graphical turn rate information was determined from a zero turn rate reference line located just above the bow of the SES. The pilot established a zero turn rate whenever the center point of the craft's bow was lined up just below the zero turn rate reference line. Variable speed lines moving from the craft's bow toward the stern gave a graphic representation of the forward velocity u (See Fig. 12). A sequence of displays during a pilot commanded control effort of left nozzle deflection and then right nozzle deflection is shown in Fig. 13, Fig. 14, and Fig. 15.

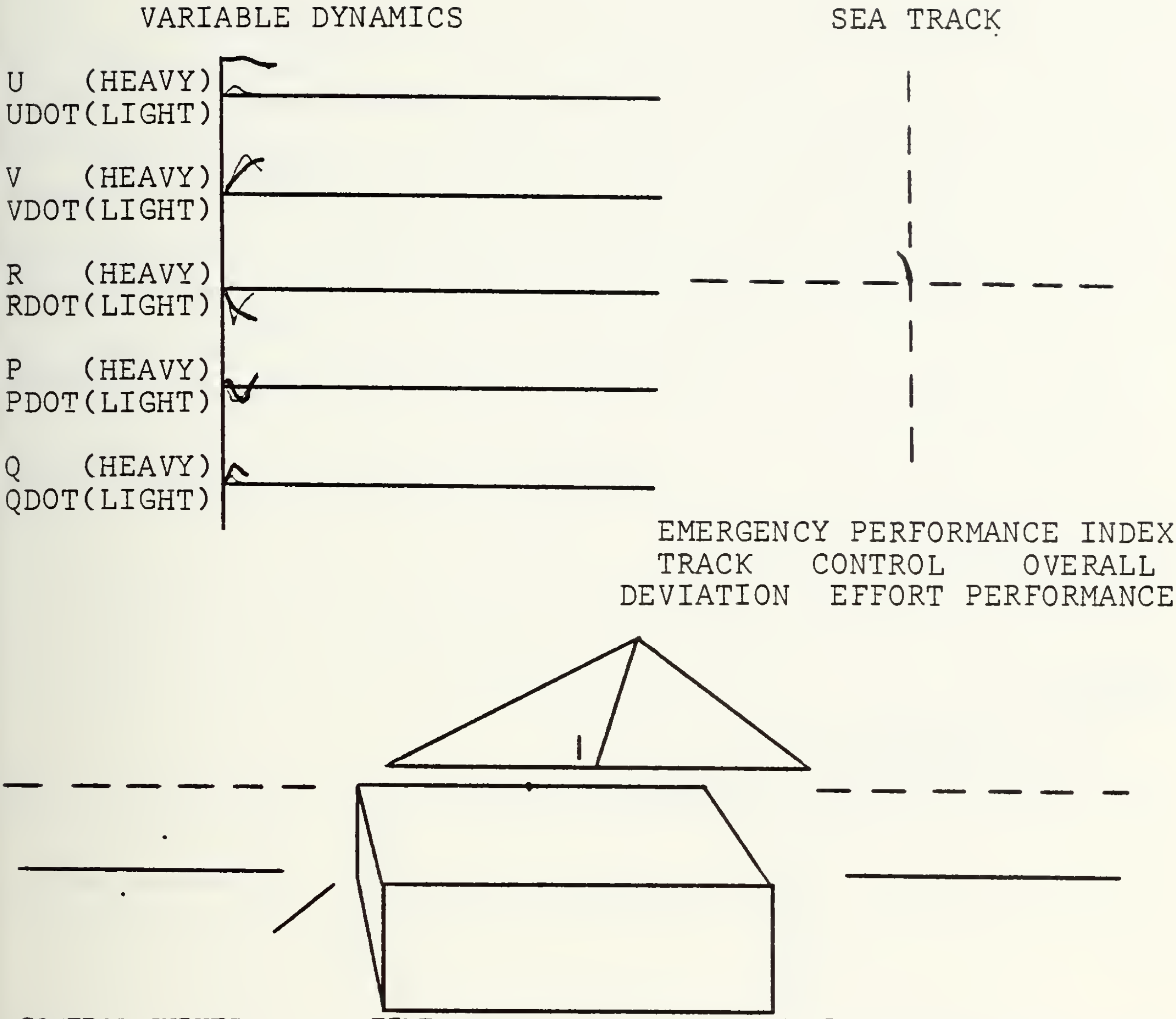
The north road positioning on the pilot's graphic display console was solved using the following assumptions:



NOTE: 1) Craft shows center of bow is left of reference line hence a left turn status.

Figure 12

SES Perspective Information
Without North Road



CONTROL INPUTS			TIME	NAVIGATIONAL DATA				
THRUST	RUDDER	DELT	ELAPSED	PITCH	HEADING/RATE	DRIFT	ROLL	SPEED
400000	.32	.084	10	.52	355/-1.0	.02	-.02	56
THRUST1	THRUST2	THRUST3	THRUST4	RUD1	RUD2	RUD3		RUD4
100000	100000	100000	100000	.32	.32	.32		.32

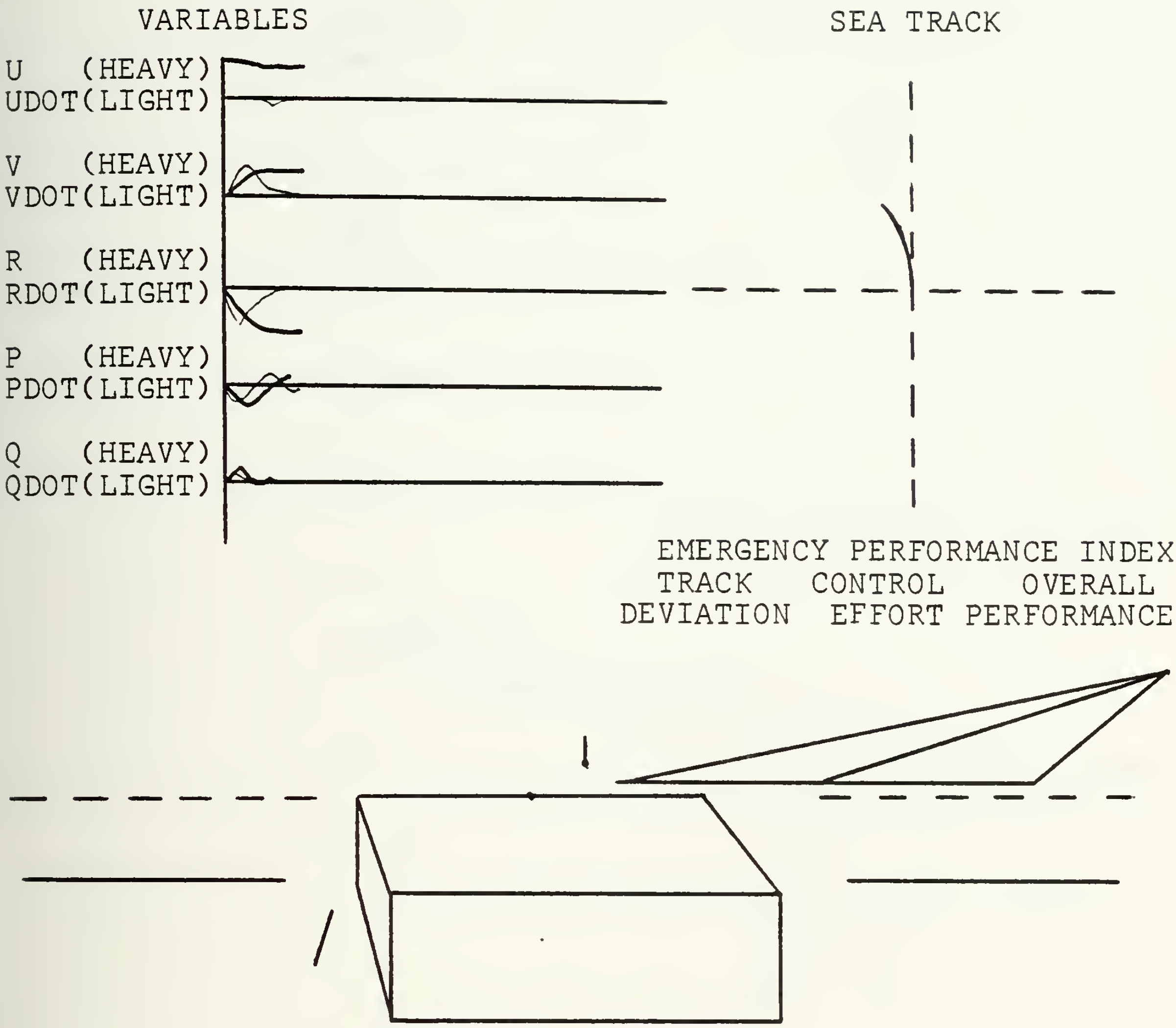
NOTE: 1) Craft has been commanded by pilot into a left turn by deflecting nozzles .32 radians (+18°). Vehicle is entering left turn and is passing through a heading of 355° with a turn rate of -1°/sec (see NAVIGATION DATA readout).

2) SEA TRACK and electronic road show vehicle has displaced slightly left of north road centerline.

3) VARIABLE DYNAMICS show velocity and acceleration time histories being generated as manuever develops.

Figure 13

SES MANUEVER (I)



CONTROL INPUTS		TIME		NAVIGATIONAL DATA				
THRUST	RUDDER	DELT	ELAPSED	PITCH	HEADING/RATE	DRIFT	ROLL	SPEED
400000	.32	.084	20.2	.58	320/1,2	.04	-.5	51
THRUST1	THRUST2	THRUST3	THRUST4	RUD1	RUD2	RUD3		RUD4
100000	100000	100000	100000	.32	.32	.32		.32

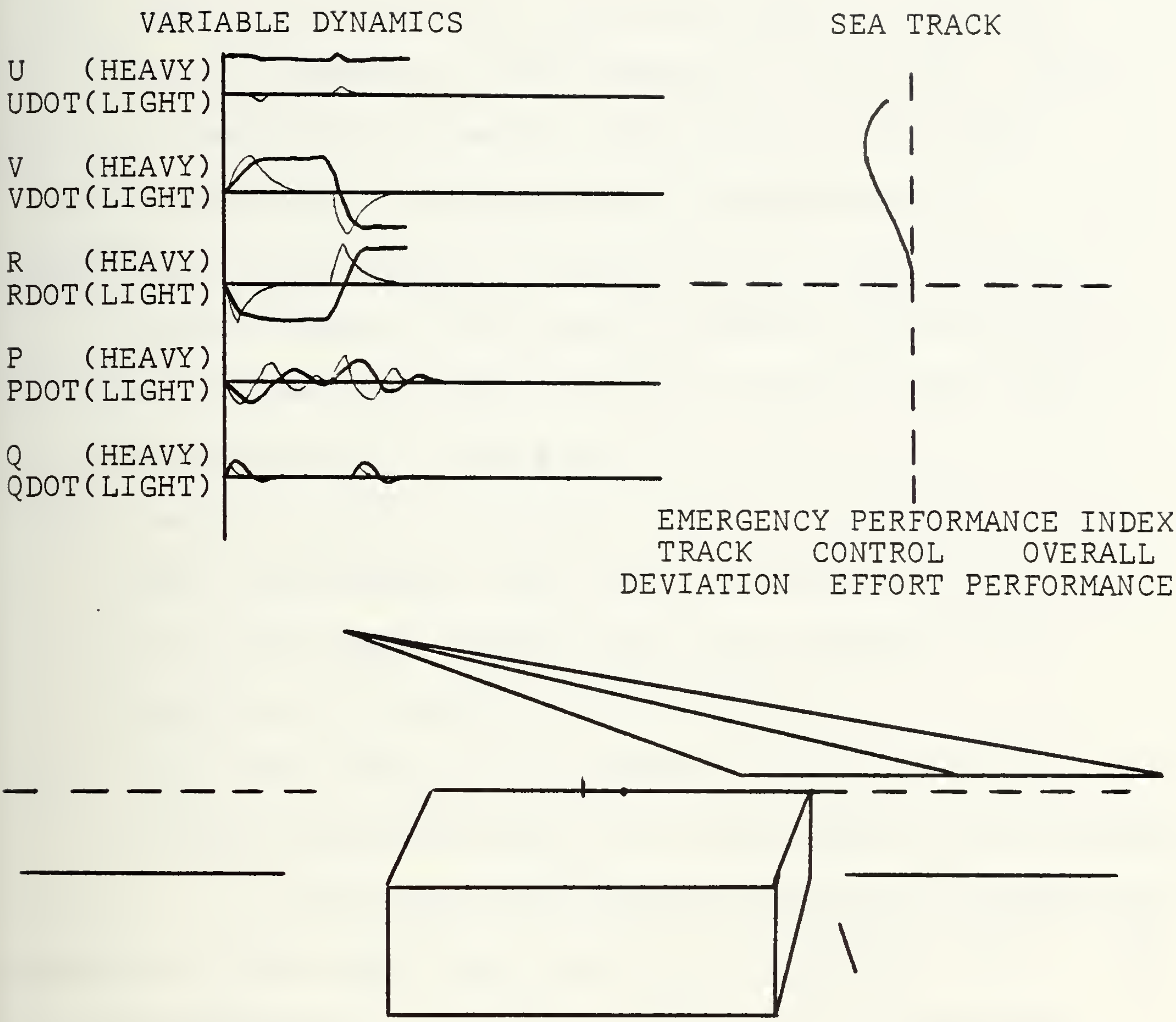
NOTE: 1) Pilot has held left nozzle deflection angle of .32 rad (+18°) and vehicle has displaced left of north road. Vehicle has heading of 320° or 40° off the north road centerline.

2) SEA TRACK confirms trajectory of craft in left turn.

3) VARIABLE DYNAMICS show +u, +v, -r, and q in steady state while roll rate p is still seeking steady state condition.

Figure 14

SES Manuever (II)



CONTROL INPUTS		TIME	
THRUST	RUDDER	DELT	ELAPSED
400000	-.5	.084	33.
THRUST1	THRUST2	THRUST3	THRUST4
100000			

NAVIGATION DATA				
PITCH	HEADING/RATE	DRIFT	ROLL	SPEED
.51	030/1.8	-.3	.8	50
RUD1	RUD2	RUD3		RUD4
-.5	-.5	-.5		-.5

NOTE: 1) Vehicle has been manuevered into right turn.
2) Vehicle heading 030° on path which will intersect north road.

Figure 15

SES Manuever (III)

1) The view of the pilot was restricted to plus and minus 60 degrees of the heading of the craft. This corresponded to the far right and far left edges of the pilot's display screen, respectively.

2) The starting edge of the road would remain fixed at a point ahead of the vehicle whenever the vehicle was positioned such that the pilot could see the road in his restricted viewing arc.

3) The road would give visual cues on the craft's left-right displacement from the road centerline and on the relationship between the north heading of the road and the heading of the vehicle.

4) The road's shape as it appeared to the pilot would be a wide path terminating at an apex point in the distance.

A non linear relationship was used to compute the position of the north road (See Fig. 16A and Fig. 16B). The Y_{NAV} displacement of points A, B, and C on the graphics display console was made a function of Y_O only. The Y_{NAV} displacement of point D on the graphics display console was made a function of Y_O and the arc tangent of the craft's heading. The X_{NAV} displacement of the points A, B, C, and D were fixed at a constant value, thus recalling Fig. 7B, the computation of the coordinates of the points A, B, C, and D was

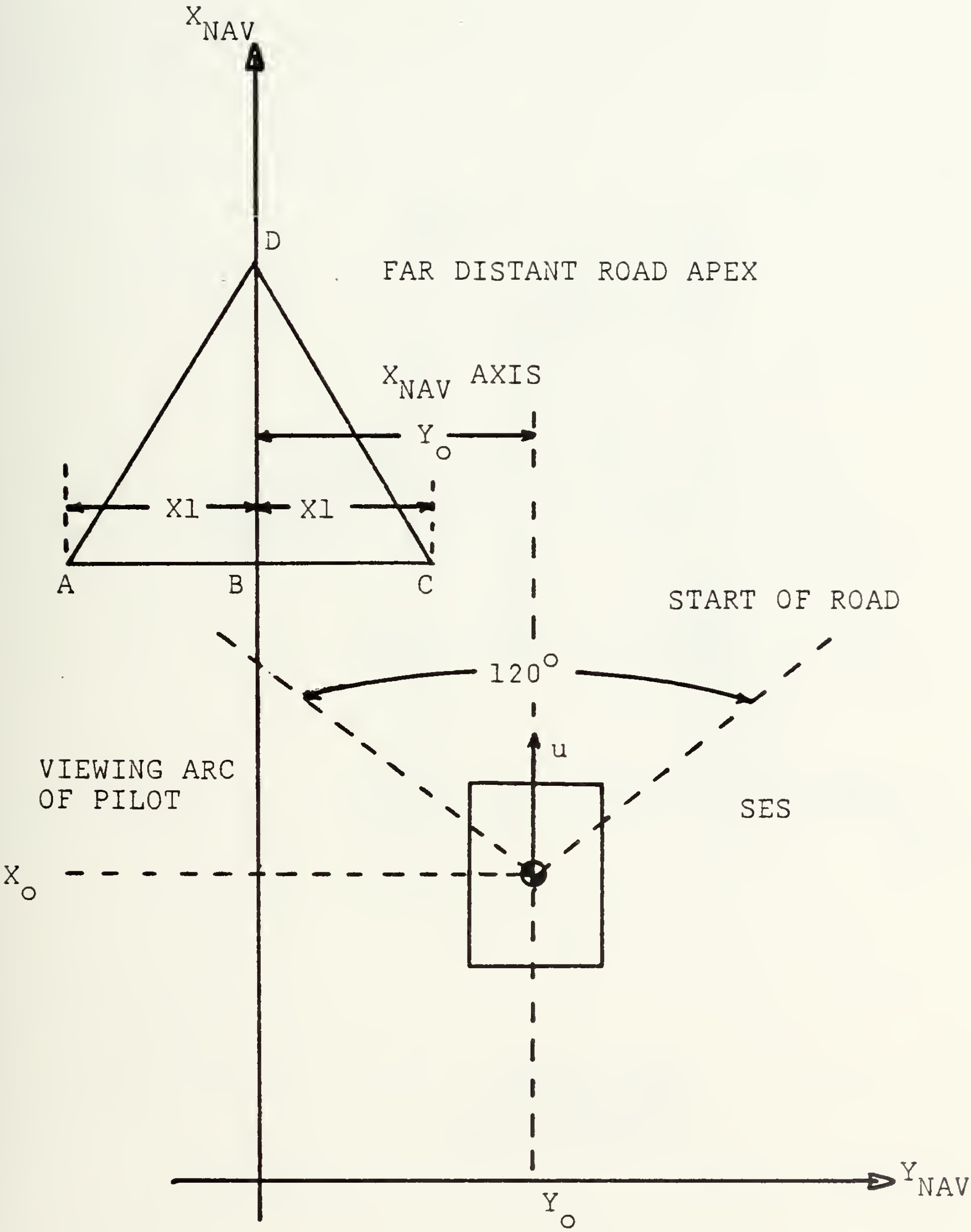


Figure 16A
North Road(I)

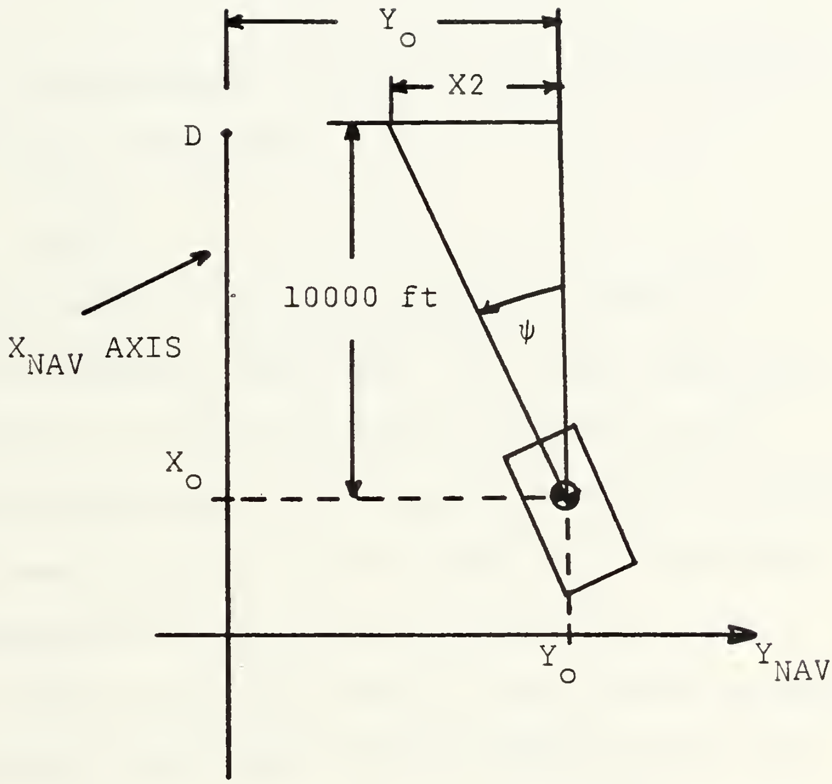


Figure 16B
North Road(II)

$$X_A = -(Y_o + X_1) \qquad Y_A = +.01$$

$$X_B = -Y_o \qquad Y_B = +.01$$

$$X_C = -(Y_o - X_1) \qquad Y_C = +.01$$

$$X_D = -(Y_o + 10000(\tan^{-1} \psi)) \quad Y_D = +.1$$

The displacement of points A, B, and C due to vehicle heading were neglected.

E. OPERATIONAL MODES

The RTS5D operated in any combination of modes as designated by the operator (See Fig. 17). Mode selection was accomplished via "on-off" switches located at the console of the XDS-9300 and at the thruster console located at the pilot's seat (See Fig. 5A and Fig. 5B). The RTS5D ran in any combination of the modes available with the "RESTART" mode taking priority over all other modes selected.

The FIXED THRUST and FIXED EFFECTOR ANGLE removed the appropriate man-generated control and utilized a constant value for T and δ respectively.

The PRINT mode allowed variables to be printed on the XDS-9300 line printer at intervals selected by the operator.

The HOLD mode put the program into a loop whereby the display screens would continue to present the last image computed until such time as the mode was deselected. At this time the program would continue.

MODE	DESCRIPTION
1	THRUST MAGNITUDE FIXED
2	THRUST MAGNITUDE VARIABLE
3	EFFECTOR DEFLECTION ANGLE FIXED
4	EFFECTOR DEFLECTION ANGLE VARIABLE
5	ACCIDENTAL FAILURE THRUST LOSS
6	ACCIDENTAL FAILURE EFFECTOR REVERSAL
7	PRINT
8	HOLD
9	RESTART ADJUST CONSTANTS IN PROGRAM
10	RESTART RETAIN CONSTANTS

Figure 17

RTS5D Modes of Operation

The RESTART mode re-initialized the program and then provided an opportunity to the operator to change the magnitude of any constant in the program. This feature allows the operator to carry out program modeling adjustments without the delays associated with re-compiling a modified digital program. Adjustments to constant magnitudes are made via a peripheral teletypewriter (TTY-35) (See Fig. 5A).

The ACCIDENTAL FAILURE FFFECTOR REVERSAL mode deflected the nozzle angle of the starboard thruster δ_{10} (See Fig. 3) to 180°. This simulated the accidental reversal of an outboard thruster at full power. Simultaneously the program computed and displayed the performance indices discussed in the Software section of this report. Exit from this mode was accomplished when the pilot had maneuvered the craft back on track. At this time the program would "hold" while displaying the last computed values of all variables. The hold condition would continue until a restart was accomplished by the operator.

The ACCIDENTAL FAILURE THRUST LOSS mode was entered by decreasing the thruster setting of any one, or combination of, thrusters (See Fig. 5B). Exit from this mode was identical to the accidental failure effector reversal described previously.

The FIXED STEP SIZE INTEGRATION mode removes the real time clock input and utilizes an operator specified time

increment on each iteration of the program loop. This is a non real time mode. The output displays and other modes are functional in this mode.

V. RTS5D RESPONSE CHARACTERISTICS

A. VALIDATION

The DBSIM5D program was utilized as the benchmark to validate the RTS5D simulation. A DBSIM5D sequence of runs was made with step effector deflection angles of 5°, 10°, 15°, 20°, 25°, and 30°. The runs were initiated with the craft at 56 knts and traveling straight ahead just prior to effector deflection. The first peak overshoot value and the steady state value attained by the variables u (surge), v (sway), r (yaw rate), ϕ (roll), and θ (pitch) were used to document error in the RTS5D. The results of the comparison are plotted as error versus effector deflection angle in Fig. 18, Fig. 19, Fig. 20, Fig. 21, and Fig. 22.

B. ACCIDENTAL FAILURE ANALYSIS

The response of the RTS5D when subjected to accidental failure mode scenarios was examined. Two primary situations involving the requirement for the pilot to react to the failure and maneuver the craft back onto its original track were analyzed. The first accidental failure situation consisted of reversing the starboard thruster T_{10} (See Fig. 3), such that δ_{10} equalled 180° while the vehicle was tracking straight ahead at 56 knts. The second situation consisted of again reversing the T_{10} thruster to δ_{10} equal 180° with the additional requirement that the failure occur while the vehicle was established in a steady state full right turn ($\delta_7=\delta_8=\delta_9=\delta_{10}=-30°$). To assume that the "worst case" accidental

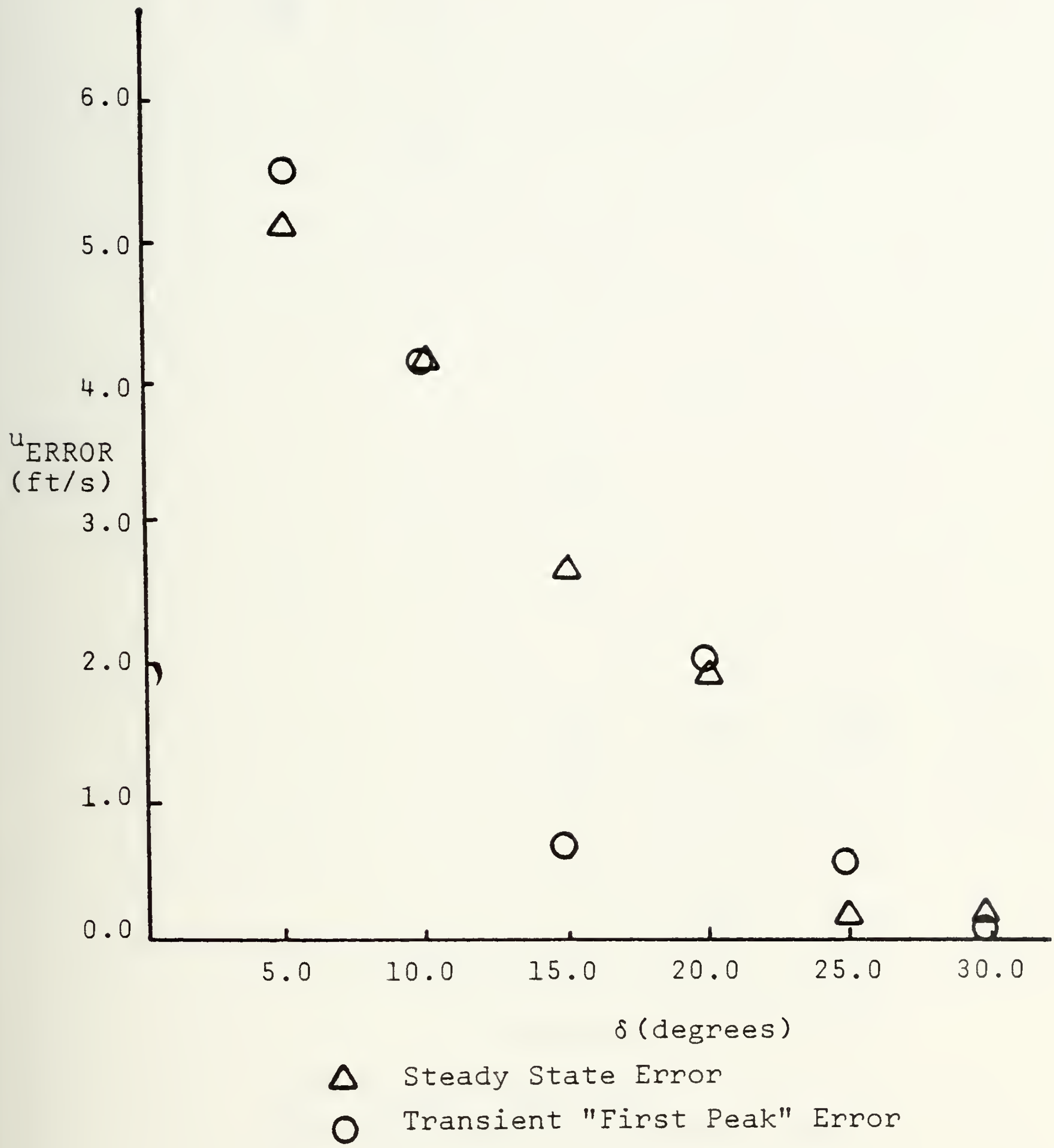


Figure 18

Effector Angle vs Surge Error

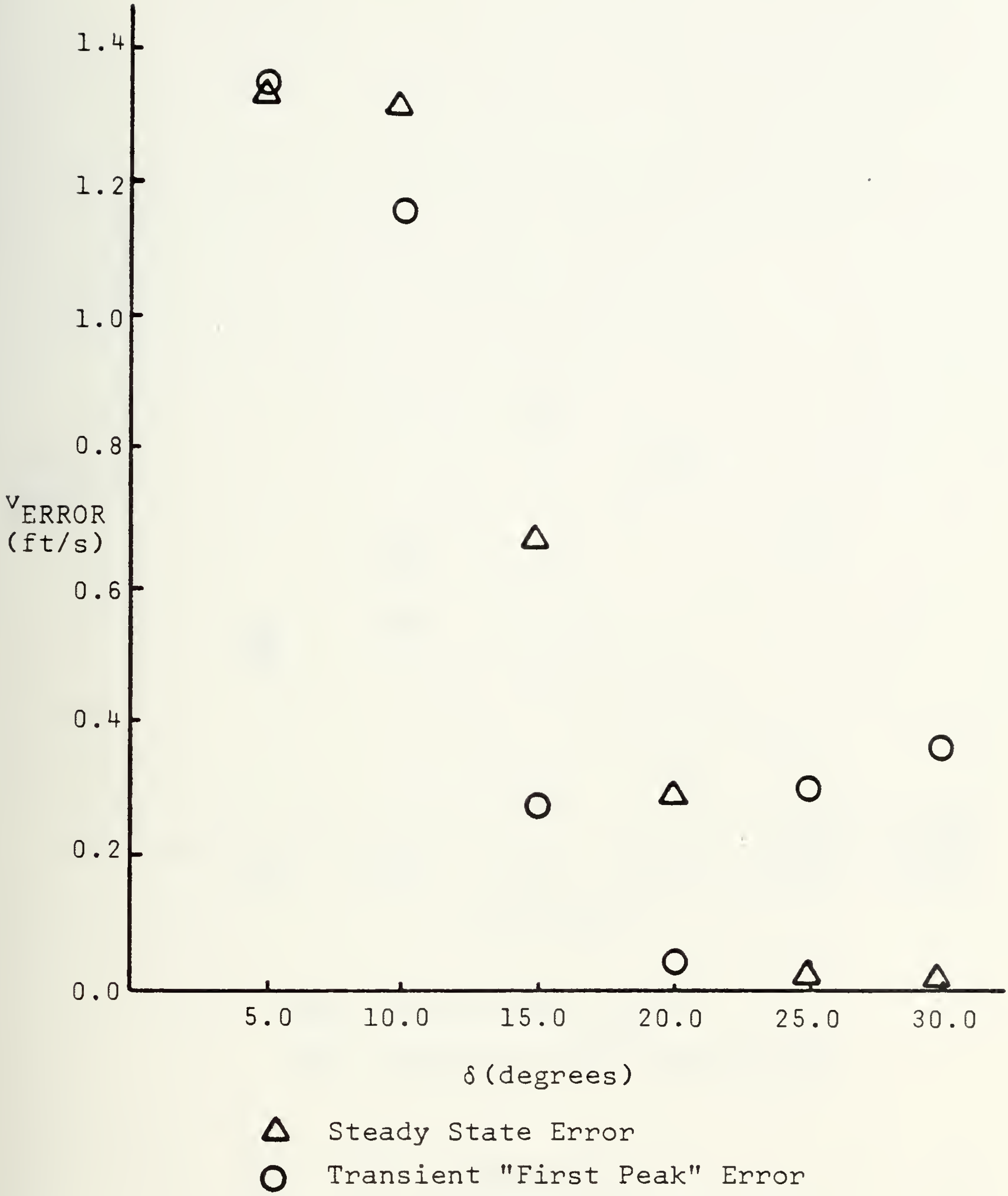
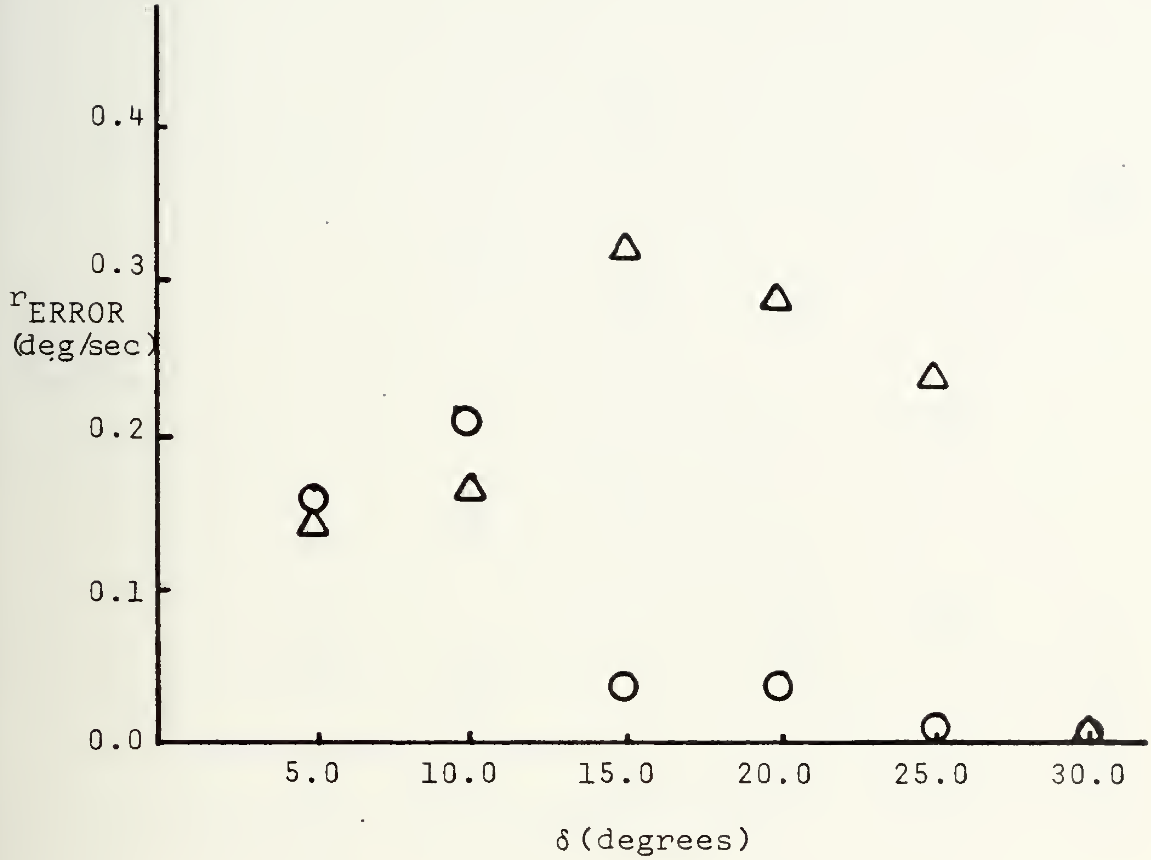


Figure 19
Effector Angle vs Sway Error



△ Steady State Error

○ Transient "First Peak" Error

Figure 20
Effector Angle vs Yaw Rate Error

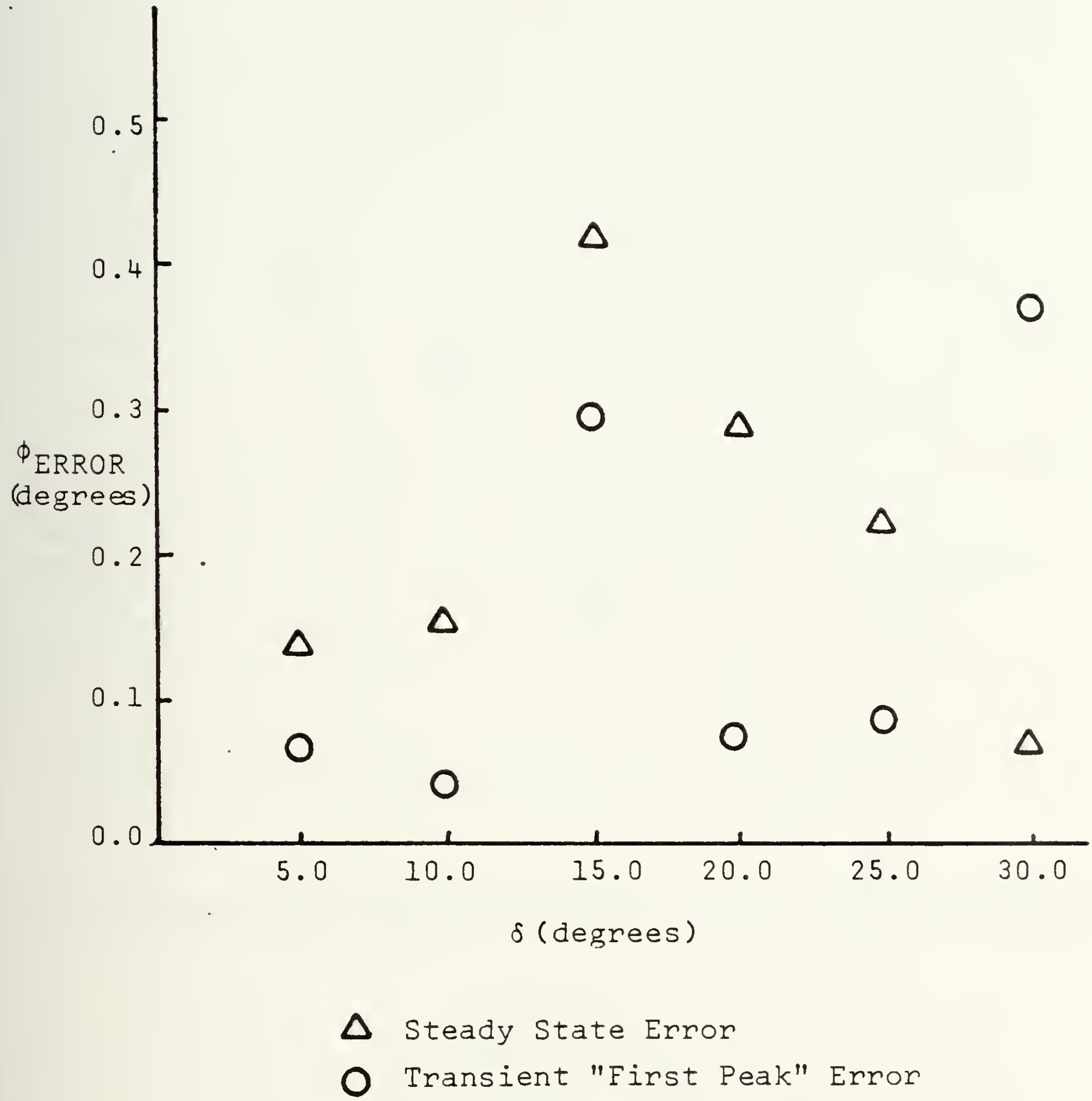


Figure 21
Effector Angle vs Roll Error

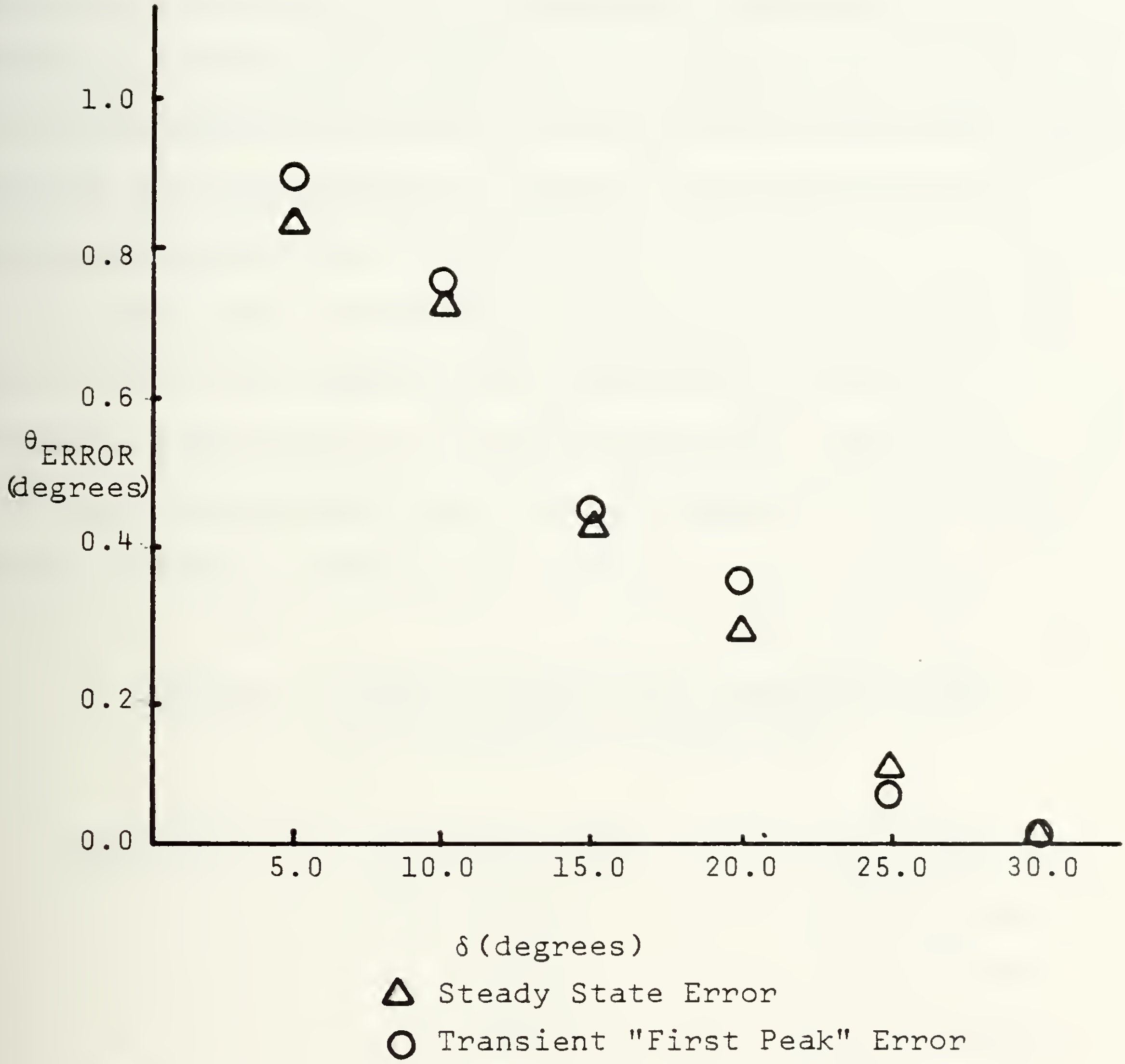


Figure 22
Effector Angle vs Pitch Error

failure scenario prevailed throughout the conduct of the two failure situations described above, the pilot was not allowed to reduce the thrust magnitudes of either the reversed effector T_{10} or the remaining thrusters T_7, T_8, and T_9. The pilot was required to recognize the failure had occurred, respond with direct real time man generated control efforts via the "joystick" (See Fig. 5A) and maneuver the vehicle back onto the "north road".

In each case the exercise was started by initializing the craft at the steady state conditions for either a straight ahead run or a -30° effector angle right turn. The appropriate steady state values for each initialization used are shown in Table IV.

TABLE IV. STEADY STATE CRAFT CONDITIONS PRIOR TO ACCIDENTAL FAILURE TEST

VARIABLE	STRAIGHT AHEAD	RIGHT TURN
u	87. ft/sec	73.99 ft/sec
v	0. ft/sec	-7.58 ft/sec
R	0. deg/sec	2.9 deg/sec
ϕ	0. deg	1.2 deg
θ	1. deg	1.19 deg
ψ	0. deg	0. deg

Termination of the exercise occurred whenever the pilot simultaneously satisfied two recovery requirements:

1) The craft's navigation coordinates must be within 50 ft of the north road centerline ($+X_{NAV}$ axis).
2) The translation velocity of the vehicle in the Y_{NAV} direction must be less than plus or minus 1 ft/sec.

It was considered that the craft was completely under control once the vehicle had fulfilled the above requirements. It is instructional to note that the recovery criteria not only required that the pilot counter any asymmetric turning moments and sway velocities induced by the failure, but that he maneuver the vehicle back to its original track. The results of a series of accidental thrust reversal tests are shown in Fig. 23A and Fig. 23B. The maximum Y_{NAV} (See Fig. 1) translation was 82.2 ft for the straight ahead run case and 571.2 ft for the right turn effector reversal case. It was observed that in both cases the craft retained the capability to maneuver back to the original track. Further, the corrective action required in both cases was simply to command the three operational effectors to the full left (+30°) position and hold that until the approach to the original track, at which time the effectors were modulated. Consequently the fastest recovery can be expected to come from the operator who is quickest to recognize that an effector reversal has occurred. The initial optimal control algorithm in both cases was full left rudder.

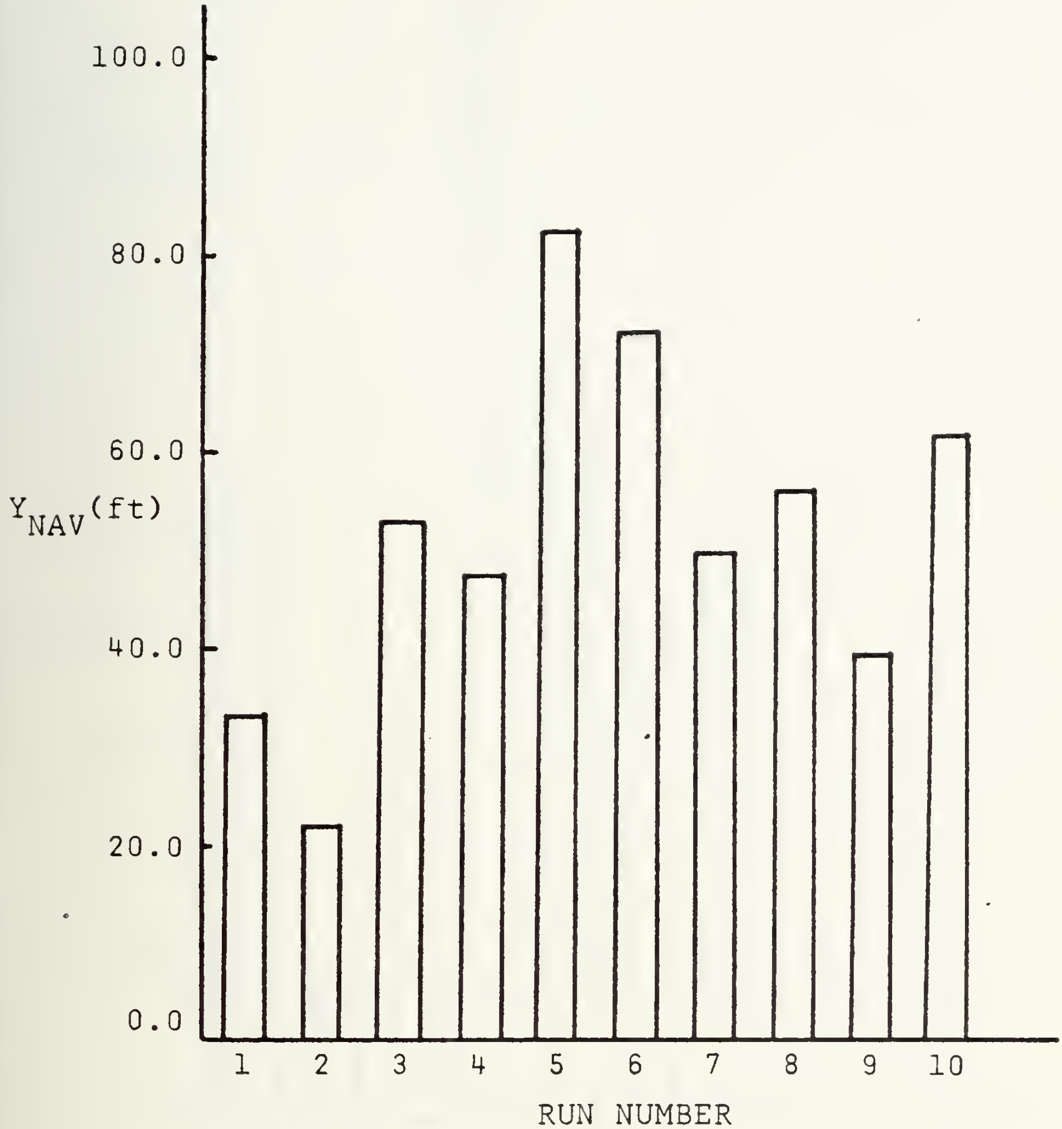


Figure 23A
Maximum Y_{NAV} Translation For
Effector Reversal, Straight Ahead Run

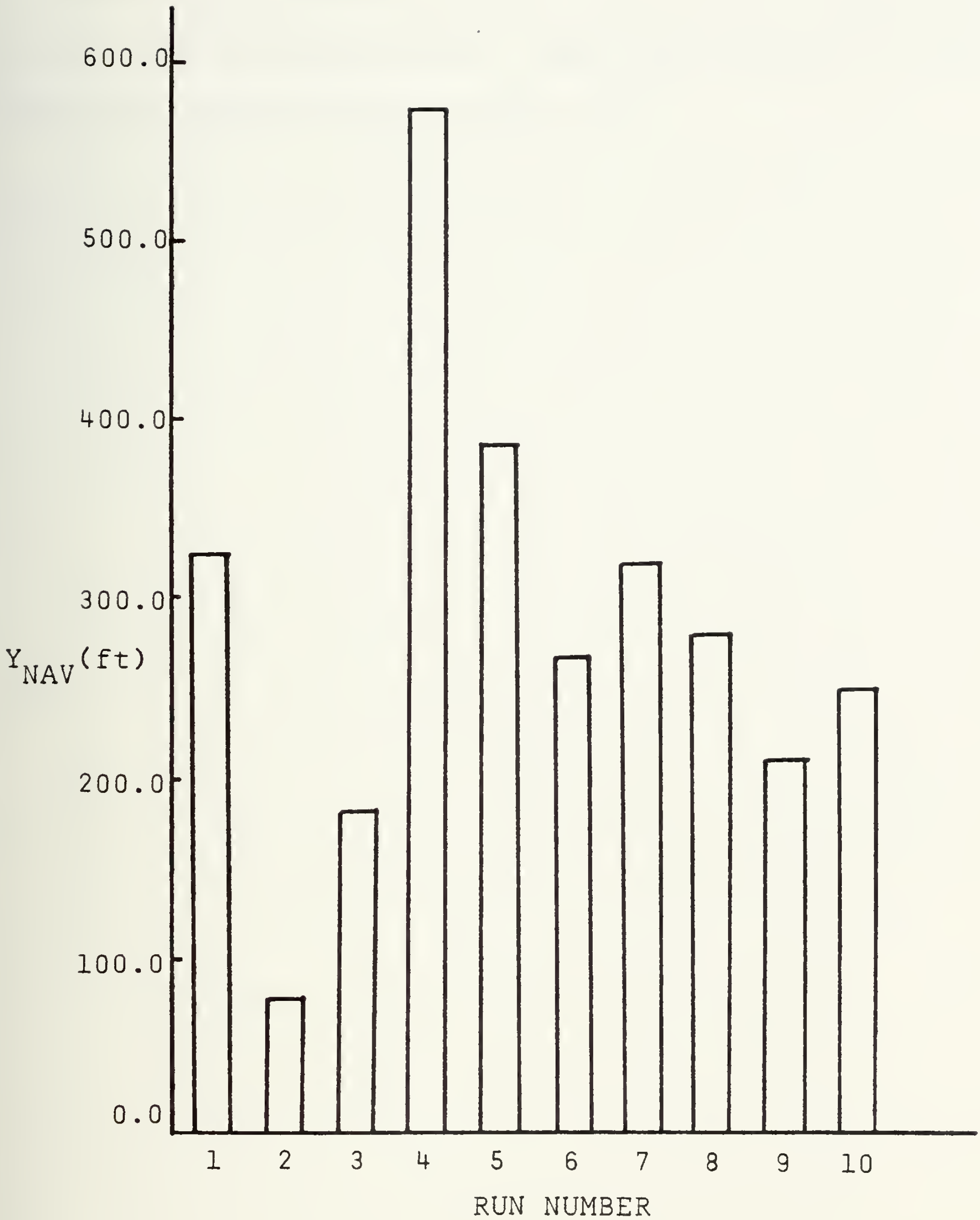


Figure 23B

Maximum Y_{NAV} Translation For
Effector Reversal, Right Turn Run

Repeated runs made with the same operator demonstrated a very steep learning curve. Rapid improvement in initial scores was demonstrated in every case.

VI. CONCLUSIONS

The ability to implement a real-time, man-controlled simulator exhibiting the steady state dynamics of the 3K-SES has been established. By requiring the right hand side of the simplified equations of motion to be restricted to a few significant terms, that portion of the iteration loop delay dedicated to computation and the AD/DA process has been reduced to an acceptable magnitude of approximately 35 msecs. The remaining portion of the total iteration time of approximately 84 msec was required for the subroutine's outputting the "alpha-numeric" presentation of the craft's states onto a graphics display console. This low value of graphics presentation time resulted from the utilization of time multiplexing of the data to be displayed. Using 100 msecs as a maximum allowable iteration time additional terms for the right hand side can be incorporated to generate a more accurate simulation. From Section III on rectuangular integration error analysis it was concluded that the RTS5D loop delay must remain less than 100 msec to remain credible. This concurs with those conclusions reached by A.L. Greensite in Ref. 9. Once the 100 msec loop delay point has been reached further expansion of the right hand side of the equations of motion would come from development of a more efficient graphics system. It was observed that while the graphics console utilization required a high cost in terms

of loop delay, the interface of realistic man-generated control efforts could be accomplished in no other available manner due to limitations on existing facilities.

Results of operational use of the RTS5D indicate the craft perspective and north road algorithm developed in Section IV-D and used to generate real time visual cues for the pilot to be extremely effective in displaying craft heading changes and vehicle displacements in Y_o.

The results in Section V-A have shown the RTS5D to be within acceptable error limitations of less than 2% on the average when compared to the DBSIM5D program in the moderate speed range of approximately 56 knts and over the full effector deflection range of plus or minus 30 degrees.

The results in Section V-B have shown that the 3K-SES is capable of being controlled satisfactorily by man during accidental failures involving both loss of thrust and effector reversals. From these results it was shown that the optimal control effort during the initial stages of an effector reversal failure was simply to apply full opposing effector angle on the remaining operational effectors. This input was held by the pilot until the craft returned to the vicinity of the original track at which time the operational effectors deflection angles had to be man-manipulated slightly to maintain track. Consequently computer assisted maneuvering controls would be somewhat helpful but do not

appear to be a hard requirement during effector reversal scenarios in the 3K-SES.

It was also concluded that the RTS5D program can be readily expanded to a full 6 degree of freedom simulator with the addition of simplified equations representing the heave motion characteristics of the 3K-SES.

VII. RECOMMENDATIONS

The following recommendations are made for future studies utilizing the RTS5D to simulate the 3K-SES:

1. The current right hand side of the craft's equations of motion require expansion to include more terms to accurately represent the 3K-SES over a wider operating range.

2. Incorporate the sixth degree of freedom, heave, into the simulation.

3. Include "sea-state" input whenever more computational time becomes available.

4. Program the propulsion plant dynamics to allow constant power as an input in lieu of constant thrust.

5. Design a dedicated micro-processor or mini-computer to assist the operator in maneuvering control with the intent of more fully examining the computer assisted controls requirement in the 3K-SES.

6. Incorporate actual 3K-SES control hardware into the RTS5D to allow more realistic "hands on" testing of not only the control hardware itself but also the operator personnel. Currently a project is being designed to test operator response under various command and failure conditions. Results will be reported in the near future.

APPENDIX A

RTS5D PROGRAM NOMENCLATURE

A11	A/D trunk 500 voltage
A21	A/D trunk 501 voltage
A22	Added mass coefficient in yaw moment equation
A31	A/D trunk 502 voltage
A33	Added mass coefficient in roll moment equation
A34	Added mass coefficient in pitch moment equation
A41	A/D trunk 503 voltage
A51	A/D trunk 504 voltage
A61	A/D trunk 505 voltage
A71	A/D trunk 506 voltage
A81	A/D trunk 507 voltage
A91	A/D trunk 510 voltage
AASTOP	performance index limit
ABSYD	$\|Y_o\|$
ABSYOD	$\|\dot{Y}_o\|$
AD	array of A/D lines
A1COA	speed line index counter
AM	mass
APITCH	pitch angle in degrees
APOTK5	voltage on pot E of thruster console
APRINT	# iterations between print execute commands
AR	analog turn rate voltage

ARATE	turn rate in degrees
ARDOT	analog $\dot{r}$ voltage
AROLL	roll angle
ASTOP	performance index J_1 limit
AU	analog velocity voltage
AUDOT	analog $\dot{u}$ voltage
AV	analog sway velocity voltage
AVDOT	analog $\dot{v}$ voltage
AX	X-Y plotter scale factor
AXD	X-Y plotter X_o coordinate
AXST	initial condition X-coordinate on sea track
AY	X-Y plotter scale factor
AYD	X-Y plotter Y_o coordinate
AYST	initial condition Y-coordinate on sea track
B101	D/A trunk T-432
B11	D/A trunk T-420
B111	D/A trunk T-433
B21	D/A trunk T-421
B31	D/A trunk T-422
B41	D/A trunk T-423
B51	D/A trunk T-424
B55	D/A trunk T-425
B61	D/A trunk T-426
B71	D/A trunk T-427
B81	D/A trunk T-430
B91	D/A trunk T-431
BETA	drift angle

BOAT	subroutine to generate perspective of SES
CDP	bow real forces lumped coefficient
CDX	drag forces lumped coefficient
CDY	sway forces lumped coefficient
CDZP	sidewall rolling moment lumped coefficient
DAL	digital to analog call
DELT	iteration loop time
DELTA	commanded iteration loop time
DTIME	v time
DTIMPLT	scaled value for v time history
DXPLOT	same as DTIMPLT
DYPLOT	scaled value for v
DYREPET	restart value for v
EFLG	error flag for TEXTO call
ETIME	r time
EYREPET	restart value for r
EXPLOT	scaled r time
EYPLOT	scaled r
F1	stern buoyancy force
F2	bow seal pitch force
F3	bow buoyancy force
FSP	port sidewall buoyancy force
FSS	starboard sidewall buoyancy force
FTIME	$\dot{v}$ time
FTIMPLT	scaled $\dot{v}$ time
FXPLOT	same as FTIMPLT
FYPLOT	scaled $\dot{v}$

FYREPET	restart value for $\dot{v}$
G	gravitational acceleration
GRAPHO	subroutine to project image (non alpha-numeric)
GTIME	$\dot{R}$ time
GTIMPLT	scaled $\dot{R}$ time
GXPLOT	same as GTIMPLT
GYPLOT	scaled $\dot{R}$
GYREPET	restart $\dot{R}$ value
HEAD	craft heading
HOLD	subroutine to freeze display/program
HTIME	$\dot{u}$ time
HTIMPLT	scaled $\dot{u}$ time
HXPLOT	same as HTIMPLT
HYBETA	analog B
HYBRID	analog flag coefficient (0 or 1)
HYPLOT	analog y axis sea track coordinate
HYPSI	analog heading (radians)
HYR	analog turn rate (rad)
HYRDOT	analog $\dot{r}$
HYREPET	analog restart y coordinate on sea track
HYSESX	analog y position of craft
HYSESY	analog x position of craft
HYU	analog u
HYUDOT	analog $\dot{u}$
HYV	analog v
HYVDOT	analog $\dot{v}$

HYXO	analog X_o
HYXODOT	analog $\dot{X}_o$
HYYO	analog Y_o
HYYODOT	analog $\dot{Y}_o$
ICO	multiplex graphics index
ICOA	speed line index
ICRAFT	vehicle designator
IER	error flag
IIPR	time history limit
IIPRD	time history limit
IIPRE	time history limit
IIPRF	time history limit
IIPRG	time history limit
IIPRH	time history limit
IIPRO	time history limit
IIPRP	time history limit
IIPRQ	time history limit
IIPRR	time history limit
IIPRS	time history limit
IIX	print counter
IJ	craft perspective counter
IJA	craft perspective counter
IJAD	time history counter
IJAE	time history counter
IJAF	time history counter
IJAG	time history counter
IJAH	time history counter

IJAO	time history counter
IJAP	time history counter
IJAQ	time history counter
IJAR	time history counter
IJAS	time history counter
IJD	time history counter
IJE	time history counter
IJF	time history counter
IJG	time history counter
IJH	time history counter
IJO	time history counter
IJP	time history counter
IJQ	time history counter
IJR	time history counter
IJS	time history counter
IL	graphics console display array
ILA1	graphics console display array
ILA2	graphics console display array
ILA3	graphics console display array
ILA4	graphics console display array
ILA5	graphics console display array
ILA6	graphics console display array
ILA7	graphics console display array
ILA8	graphics console display array
ILA9	graphics console display array
ILA10	graphics console display array

ILA11	graphics console display array
ILA12	graphics console display array
ILA13	graphics console display array
ILA14	graphics console display array
ILA15	graphics console display array
ILA16	graphics console display array
ILA17	graphics console display array
ILA18	graphics console display array
ILA19	graphics console display array
ILA20	graphics console display array
ILA21	graphics console display array
ILA22	graphics console display array
ILA23	graphics console display array
ILA24	graphics console display array
ILA25	graphics console display array
ILA26	graphics console display array
ILA27	graphics console display array
ILA28	graphics console display array
ILA29	graphics console display array
ILAP	graphics console display array
IN	failure mode counter
IPACK	subroutine to load display arrays
IPERFT	performance index
IPLOT	graphics console display array
IPR	time history counter limit
IPRD	time history counter limit

IPRE	time history counter limit
IPRF	time history counter limit
IPRG	time history counter limit
IPRH	time history counter limit
IPRO	time history counter limit
IPRP	time history counter limit
IPRQ	time history counter limit
IPRR	time history counter limit
IPRS	time history counter limit
ISWA	display array
ITEXT	subroutine to display alpha-numeric data
IVEIWA	craft perspective/north road array
IVIEWAA	sea track array
IVIEWB	sea track axis array
IVIEWC	u time history array
IVIEWD	v time history array
IVIEWE	r time history array
IVIEWF	$\dot{v}$ time history array
IVIEWG	$\dot{r}$ time history array
IVIEWH	$\dot{u}$ time history array
IVIEWO	$\dot{p}$ time history array
IVIEWP	$\dot{q}$ time history array
IVIEWQ	p time history array
IVIEWR	q time history array
IVIEWS	analog sea track array
IVIEWZ	time history axis array

IX	x axis moment of inertia
J	time history counter
JD	time history counter
JE	time history counter
JF	time history counter
JG	time history counter
JH	time history counter
JO	time history counter
JP	time history counter
JQ	time history counter
JR	time history counter
JS	time history counter
L3	lever arm for buoyancy force
LD	draft of sidewall
LP	side thrust component lever arm
LZ	lever arm for sway drag force
N	clock interrupt count
NAD	AD trunk line array
NDA	DA trunk line array
OO	lever arm for thrust side component
OTIME	$\dot{p}$ time
OTIMPLT	scaled $\dot{p}$ time
OXPLOT	same as OTIMPLT
OYPLOT	scaled $\dot{p}$
OYREPET	restart $\dot{p}$ value
P	roll rate
PB	plenum pressure

PDOT	roll acceleration
PDOTM	max roll acceleration
PHI	roll angle
PMAX	max roll rate
POTK	thruster console array
POTS	control stick array
POTSA	function box array
PSI	craft heading
PTIME	$\dot{q}$ time
PTIMPLT	scaled $\dot{q}$ time
PXPLOT	same as PTIMPLT
PYPLOT	scaled $\dot{q}$
PYREPET	restart $\dot{q}$ value
Q	pitch rate
QDOT	pitch acceleration
QDOTM	max $\dot{q}$
QMAX	max q
QTIME	p time
QTIMPLT	scaled p time
QXPLOT	same as QTIMPLT
QYPLOT	scaled p value
QYREPET	restart p value
R	craft turn rate r
RDOT	$\dot{r}$
RDOTM	max $\dot{r}$
READCLOCK	subroutine to sample real time clock

RESET	subroutine to reset analog computer
RHO	density of water
RMAX	maximum yaw rate of craft
RPERF1	J_1
RPERF2	J_2
RPERF3	J_3
RTIME	q time
RTIMPLT	scaled q time
RX	I_X
RXPLOT	same as RTIMPLT
RY	I_Y
RYPLOT	scaled q value
RYREPET	restart q value
RZ	I_Z
S1	T_9 lever arm
S2	T_{10} lever arm
S3	T_8 lever arm
S4	T_7 lever arm
SESX	scaled X_o
SESY	scaled Y_o
SF1	scale factor in time history arrays
SF14	scale factor in time history arrays
SLIP	Beta influence coefficient
SPDLIN	speedline lower screen limit
SPEED	craft's velocity in knts
STARTCLOCK	subroutine to start clock

STIME	analog time
STIMPLT	scaled analog time
SXPLOT	HYSESX
SYPLOT	HYSESY
SXREPET	restart value for HYSESX
SYREPET	restart value for HYSESY
T	total thrust
T10	thruster number 4
T10C	forward vector component of T10
T10MAX	max T10 value
T10S	side vector component of T10
T7	thruster number 1
T7C	forward vector component of T7
T7MAX	max T7 value
T7S	side vector component of T7
T8	thruster number 2
T8C	forward vector components of T8
T8MAX	max value of T8
T8S	side vector component of T8
T9	thruster number 3
T9C	forward vector component of T9
T9MAX	max value of T9
T9S	side vector component of T9
TEXTO	subroutine to display alpha-numeric data
TFORW	summation of forward thrust components

THETA	pitch angle
TIM	u time
TIME	time
TIMPLT	scaled TIM
TINT	real clock interrupt frequency
TITLE	program name display array
TITLE0	program name display array
TITLE1	program name display array
TMAX	max T
TSIDE	sum of side thrust components
TYAW	sum of yaw moment
U	forward velocity (surge)
UDOT	$\dot{u}$
UDOTM	max $\dot{u}$
UMAX	max u
UMUL	scale factor in speedline generation
V	side velocity (sway)
VCD	thruster console pot array
VDOT	$\dot{v}$
VDOTM	max $\dot{v}$
VMAX	max v
VS	total velocity
W	heave velocity
WDOT	heave acceleration
WE	width of bow seal
WRITECLOCK	subroutine to assign clock interrupts value to variable

WW	lever arm for sway drag forces
X	Y_o
XO	X_o
XODOT	$\dot{X}_o$
XDX	boat perspective scale factor
XGX	boat perspective scale factor
XPLOT	display array
XREPET	reset value for SESX
XST	scale factor for analog plot
YO	Y_o
YODOT	$\dot{Y}_o$
YHIGH	point D vertical position constant
YPLOT	display array
YREPET	reset value for SESY
YST	scale factor for analog plot
Z	operational effector angle
Z10	effector number four angle
Z2	upper effector angle limitation
Z3	lower effector angle limitation
Z4	zero angle upper limit
Z5	zero angle lower limit
Z7	effector number one angle
Z7I	modified effector number one angle
Z8	effector number two angle
Z8I	modified effector number two angle
Z9	effector number three angle

Z9I	modified effector number three angle
ZAB	scale factor for Z10
ZI	modified Z
ZZZZ	fixed step effector angle input

APPENDIX B

RTS5D Computer Program Listing

```
      DIMENSION ITEXT(29),ILAP(24),POTS(24),POTK(24),IVIEWA(75)
     C,IL(29),TITLE(24),AD(10),AA(12),ILA1(24),ILA2(24),
     CILA4(24),ILA5(24),ILA6(24),ILA7(24),ILA8(24),IPLOT(29),ILA3(24),
     CIVIEWB(70),ISWA(40),FCTSA(40),ILA9(24),IVIEWC(205),IVIEWZ(70),
     CTITLE0(24),TITLE1(24),IVIEWD(205),IVIEWE(205),IVIEWF(205),
     CIVIEWG(205),IVIEWH(205),ILA10(24),ILA11(24),ILA12(24),ILA13(24),
     CILA14(24),ILA15(24),ILA16(24),IVIEWAA(500),IVIEWO(205),
     CIVIEWP(205),IVIEWQ(205),IVIEWR(205),IVIEWS(205),
     CILA17(25),ILA18(24),ILA19(24),ILA20(24),ILA21(24),ILA22(24),
     CILA23(24),ILA24(24),ILA25(24),ILA26(24),ILA27(24),ILA28(24),
     CILA29(24),H(50),IIPR(20),IPR(20),J(20),IJ(20),IJA(20),XREPET(20),
     CYREPET(20),TIM(20),TIMPLT(20),XPLOT(20),YPLOT(20)
      COMMON R,THETA,PHI,L,LMAX,XGX,XDX,UMUL,YO,HEAD,YHIGH,APCTK5,
     CDELT,Z,SPELIN,AICCA,ICCA,POTK
      EQUIVALENCE (AD(1),A11),(AD(2),A21),(AD(3),A31),(AA(1),B11),
     C(AA(2),B21),(AA(3),B31),(AA(4),B41),(AA(5),B51),(AA(6),B55),
     C(AA(7),B61),(AA(8),B71),(AA(9),B81),(AA(10),B91),(AA(11),B101),
     C(AA(12),B111),(AD(4),A41),(AD(5),A51),(AD(6),A61),
     C(AD(7),A71),(AD(8),A81),(AD(9),A91),(AD(10),A101)
      NAMELIST
 1111 CONTINUE
      CALL RESET(1000)
      CALL HOLD
      CALL DAL(.0,.0,.0,.0,.0,.0,.0,.0,.0,.0,.0,.0)
      CALL DTINIT(1,ITEXT,29,IER)
      CALL DGINIT(1,ISWA,40,IER)
      CALL DTINIT(2,IPLOT,29,IER)
      CALL DGINIT(2,IL,29,IER)
      ENCODE(96,1,TITLE0)
    1 FORMAT('          REAL TIME           ')
      CALL TEXTC(1,TITLE0,24,10,1,3,3,IER)
      IF(IER.NE.0)OUTPUT(101) IER, '    1          '
      ENCODE(96,2,TITLE1)
    2 FORMAT('    MOTION ANALYSIS PROGRAM    ')
      CALL TEXTC(1,TITLE1,24,13,1,3,3,IER)
      IF(IER.NE.0)OUTPUT(101) IER, '    2          '
      I=1000000
      CALL DELAY
      V=0.
      W=0.
      R=0.
```

```
      P=C.
      Q=0.
      XO=0.
      YO=0.
      ZZ=0.
      PSI=0.
      PHI=C.
      THETA=0.
      UDCT=0.
      VDCT=C.
      WDCT=0.
      RDCT=C.
      PDCT=C.
      QDCT=0.
      TIME=0.
      DO 3131 I=1,50
 3131 H(I)=0.
      IF((PCTK(2).GT..25).AND.(PCTK(2).LT..5))GO TC 21
      OUTPUT(101)'  ICRAFT,,,,XR3=3..,,,,,,   3K-TCN=3000  '
      INPUT(101)
   21 CONTINUE
      IF(ICRAFT.EQ.3)GO TC 3000
      GC TC 3001
 3000 CONTINUE
      UMAX=33.8
      VMAX=1.50
      RMAX=0.32
      UDCTM=C.5
      VDCTM=1.2
      RDCTM=C.12
      L=33.8
      TMAX=379.
      T7MAX=94.75
      T8MAX=94.75
      T9MAX=94.75
      T10MAX=94.75
      AM=188.
      CDX=C.3396
      CDY=1064.
      CO=11.
      WW=1.28
      FZ=10500.
      AU=.45
      AUDCT=.45
      AV=.02C
      AVDOT=.C2C
      AR=0.005
      ARDCT=C.CC5
```

```
      AXST=5000.
      AYST=5000.
      S1=5.
      S2=5.
      S3=5.
      S4=5.
      A22=1.
      GO TO 3002
 3001 CONTINUE
      LMAX=099.
      VMAX=10.
      RMAX=0.125
      FMAX=.04
      GMAX=.0015
      LDCTM=1.8
      VDCTM=3.6
      RDCTM=0.040
      FDCTM=.1
      GDCTM=.001
      A33=294118.
      A34=588236.
      SLIP=0.
      LRF=50.
      LR=30.
      LZ=30.
      L3=100.
      LD=3.
      WE=100.
      FB=240.
      LP=5.
      CDP=9.174
      CDZ=1000000.
      CDZP=13336.7
      FX=18220000.
      RY=81589900.
      L=087.
      TMAX=392000.
      T7MAX=98000.
      T8MAX=98000.
      T9MAX=98000.
      T10MAX=98000.
      AM=170364.
      CDX=50.33
      CDY=14544.4
      A22=1.
      CO=110.8
      WW=26.04
      RZ=95270000.
```

```
      AU=2.
      AUDOT=2.
      AV=0.2
      AVDOT=0.2
      AR=0.005
      ARDOT=0.005
      AYST=20000.
      AXST=20000.
      S1=50.
      S2=50.
      S3=50.
      S4=50.
3002  CONTINUE
      ZZZZ=0.5263
      DELTA=0.084
      SFZZX=-.1
      Z1=2.
      Z2=0.5263
      Z3=-0.5263
      Z4=0.03
      Z5=-0.03
      Z6=-.05
      Z11=3.1416
      Z12=0.2
      XST=0.75
      YST=0.75
      XREPET(1)=XST
      YREPET(1)=YST
      SXREPET=XST
      SYREPET=YST
      AX=5000.
      AY=5000.
      V=0.
      R=0.
      W=0.
      X0=0.
      Y0=0.
      HYY0=0.
      HYX0=0.
      ZZ=0.
      PHI=0.
      THETA=0.02
      PSI=0.
      UDOT=0.
      VDOT=0.
      RDOT=0.
      WDOT=0.
      PDOT=0.
```

```
GDCT=0.
SF1=12.
SF14=0.9
JO=0
JP=0
JQ=0
JR=0
JS=0
CTIME=0.
FTIME=0.
GTIME=0.
RTIME=0.
STIME=0.
CYREFET=.3
FYREFET=.1
GYREFET=.3
RYREFET=.1
JD=0
JE=0
JF=0
JG=0
JH=0
DTIME=0.0
ETIME=0.0
FTIME=0.0
GTIME=0.0
HTIME=0.0
DYREFET=0.7
EYREFET=0.5
FYREFET=0.7
GYREFET=0.5
HYREFET=0.9
UMLL=4.0
IPR(1)=230
IPR(2)=110
IPRD=110
IPRE=110
IPRF=110
IPRG=110
IPRH=110
IPRO=110
IPRP=110
IPRQ=110
IPRR=110
IPRS=110
ICCA=0
ICC=1
AICCA=10.
```

```
      NDA=12
      NAC=10
      X=-1C0.
      TIME=0.0
      AAAA=0.
      F=C.C
      G=0.0
      RHC=2.0
      G=32.2
      AXO=25.
      AYC=25.
      TINT=1CCC.
      HYBRIC=0.
      XGX=24.
      XDX=24.
      ZAE=-3.1416
      YHIGH=C.1
      AASTCF=1.C
      ASTCF=C5C.
      RPERF1=C.
      RPERF2=0.
      RPERF3=C.
      IPERFT=C.
      IN=0
      J(1)=C
      J(2)=0
      TIM(2)=0.
      YREPET(2)=C.9
      APRINT=5.
      IF((PCTK(2).GT..25).AND.(POTK(2).LT..5))GO TC 22
1112  CUTPUT(1C1)'CHANGES,,*+ C/R'
      INPUT(1C1)
      IF(IER.NE.C)CUTPUT(1C1) IER, '    3          '
22    CONTINUE
65C   CALL CTINIT(1,ITEXT,29,IER)
      CALL DGINIT(1,ISWA,4C,IER)
      CALL CTINIT(2,IPLOT,29,IER)
      CALL DGINIT(2,IL,29,IER)
      IVIEWB(1)=IHEAD(1,10)
      IVIEWE(2)=IPACK(0.0,1.0,0)
      IVIEWB(3)=IPACK(0.0,C.5,C)
      IVIEWE(4)=IPACK(-1.C,0.0,0)
      IVIEWE(5)=IPACK(-0.5,0.0,1)
      IVIEWE(6)=IPACK(0.5,C.C,C)
      IVIEWE(7)=IPACK(1.0,0.0,1)
      IVIEWE(8)=IPACK(0.75,1.0,0)
      IVIEWE(9)=IPACK(0.75,0.5,1)
      IVIEWE(10)=IPACK(0.5,0.75,0)
```

```
      IVIEWB(11)=IPACK(1.0,0.75,1)
      IVIEWB(12)=0
      CALL GRAPHC(1,IVIEWB,12,1,IER)
      IVIEWZ(1)=IHEAD(0,10)
      IVIEWZ(2)=IPACK(-.9,0.05,0)
      IVIEWZ(3)=IPACK(-.9,.99,1)
      IVIEWZ(4)=IPACK(-.9,.9,0)
      IVIEWZ(5)=IPACK(0.0,.9,1)
      IVIEWZ(6)=IPACK(-.9,.7,0)
      IVIEWZ(7)=IPACK(0.,.7,1)
      IVIEWZ(8)=IPACK(-.9,.5,0)
      IVIEWZ(9)=IPACK(.0,.5,1)
      IVIEWZ(10)=IPACK(-.9,.3,0)
      IVIEWZ(11)=IPACK(.0,.3,1)
      IVIEWZ(12)=IPACK(-.9,.1,0)
      IVIEWZ(13)=IPACK(0.,.1,1)
      IVIEWZ(14)=IPACK(0.0,0.01,0)
      IVIEWZ(15)=IPACK(0.0,0.01,1)
      IVIEWZ(16)=IPACK(0.0,0.03,1)
      IVIEWZ(17)=0
      CALL GRAPHC(1,IVIEWZ,17,2,IER)
  501 FORMAT('PRESENT',2F10.2,'          ',F4.2,'   ',F8.2,'           ',F6.3,
     C'         ',F6.3)
  525 FORMAT('MAXIMUM',2F10.2,'          ',F4.2,'   ',F8.2,'           ',F6.3,
     C'         ',F6.3)
  526 FORMAT('MINIMUM',2F10.2,'          ',F4.2,'   ',F8.2,'           ',F6.3,
     C'         ',F6.3)
  507 FORMAT('PRESENT',F8.2,F10.3,'         ',F4.2,' ',F8.3,'               ',
     CF5.3,'       ',F5.3)
  527 FORMAT('MAXIMUM',F8.2,F10.3,'         ',F4.2,' ',F8.3,'               ',
     CF5.3,'       ',F5.3)
  528 FORMAT('MINIMUM',F8.2,F10.3,'         ',F4.2,' ',F8.3,'               ',
     CF5.3,'       ',F5.3)
  511 FORMAT('PRESENT',F7.3,F10.3,'   ',F10.2,'   ',F8.3,'      ',F6.3,'
     C    ',F6.3)
  529 FORMAT('MAXIMUM',F7.3,F10.3,'   ',F10.2,'   ',F8.3,'      ',F6.3,'
     C    ',F6.3)
  530 FORMAT('MINIMUM',F7.3,F10.3,'   ',F10.2,'   ',F8.3,'      ',F6.3,'
     C    ',F6.3)
  516 FORMAT(F8.0,' ',F4.2,'          ',F4.3,'   ',F6.2,'        ',F5.3,'
     C ',F7.2,'/',F5.2,'  ',F5.2,'  ',F5.2,'   ',F5.2)
      ENCODE(96,500,ILA1)
  500 FORMAT('                              POSITION')
      CALL TEXTC(2,ILA1,24,7,4,1,3,IER)
      ENCODE(96,502,ILA2)
  502 FORMAT('         X          Y          ZZ         PSI
     CPHI         THETA')
      CALL TEXTC(2,ILA2,24,8,4,1,3,IER)
```

```
      ENCODE(96,505,ILA3)
 505  FORMAT('                                          VELOCITY')
      CALL TEXTC(2,ILA3,24,15,4,1,3,IER)
      ENCODE(96,517,ILA4)
 517  FORMAT('              U           V           W           R
     CP              Q')
      CALL TEXTC(2,ILA4,24,16,4,1,3,IER)
      ENCODE(96,509,ILA5)
 509  FORMAT('                                      ACCELERATION')
      CALL TEXTC(2,ILA5,24,23,4,1,3,IER)
      ENCODE(96,512,ILA6)
 512  FORMAT('              UDOT        VDOT        WDOT        RDOT        PD
     COT          QDOT')
      CALL TEXTC(2,ILA6,24,24,4,1,3,IER)
      ENCODE(96,514,ILA7)
 514  FORMAT(' CONTROL INPUTS             TIME                              NAV
     CIGATION DATA')
      CALL TEXTC(1,ILA7,24,32,1,1,3,IER)
      ENCODE(96,515,ILA8)
 515  FORMAT('THRUST   RUDDER        DELT   ELAPSED      PITCH   HEADING/R
     CATE   DRIFT   ROLL    SPEED')
      CALL TEXTC(1,ILA8,24,33,1,1,3,IER)
      IF(IER.NE.0)OUTPUT(101) IER, '   12        '
      ENCODE(96,532,ILA9)
 532  FORMAT('THRUST 1   THRUST 2   THRUST 3   THRUST 4          RUD 1
     C RUD 2    RUD 3    RUD 4    ')
      CALL TEXTC(1,ILA9,24,36,1,1,3,IER)
      IF(IER.NE.0)OUTPUT(101) IER, '   13        '
      ENCODE(96,2050,ILA10)
 2050 FORMAT('        VARIABLE DYNAMICS
     CSEA TRACK')
      CALL TEXTC(1,ILA10,24,3,15,1,3,IER)
      ENCODE(96,2051,ILA11)
 2051 FORMAT(' U   (HEAVY)')
      CALL TEXTC(1,ILA11,24,5,1,1,3,IER)
      ENCODE(96,2054,ILA14)
 2054 FORMAT('UDOT(LIGHT)')
      CALL TEXTC(1,ILA14,24,6,1,1,3,IER)
      ENCODE(96,2052,ILA12)
 2052 FORMAT(' V   (HEAVY)')
      CALL TEXTC(1,ILA12,24,9,1,1,3,IER)
      ENCODE(96,2055,ILA15)
 2055 FORMAT('VDOT(LIGHT)')
      CALL TEXTC(1,ILA15,24,10,1,1,3,IER)
      ENCODE(96,2053,ILA13)
 2053 FORMAT(' R   (HEAVY)')
      CALL TEXTC(1,ILA13,24,12,1,1,3,IER)
      ENCODE(96,2056,ILA16)
```

```
2056 FORMAT('FOOT(LIGHT)')
     CALL TEXTC(1,ILA16,24,13,1,1,3,IER)
     ENCODE(96,2057,ILA17)
2057 FORMAT('                                        ZERO PITCH')
     CALL TEXTC(1,ILA17,24,21,45,1,3,IER)
     ENCODE(96,2058,ILA18)
2058 FORMAT('DATUM LINE')
     CALL TEXTC(1,ILA18,24,22,84,1,3,IER)
     ENCODE(96,2059,ILA19)
2059 FORMAT(' VARIABLE STATUS')
     CALL TEXTC(2,ILA19,24,4,28,2,3,IER)
     IF(IER.NE.0)OUTPUT(101) IER, '   6         '
     ENCODE(96,3010,ILA21)
3010 FORMAT(' F   (HEAVY)')
     CALL TEXTC(1,ILA21,24,16,1,1,3,IER)
     ENCODE(96,3011,ILA22)
3011 FORMAT('FOOT(LIGHT)')
     CALL TEXTC(1,ILA22,24,17,1,1,3,IER)
     ENCODE(96,3012,ILA23)
3012 FORMAT(' G   (HEAVY)')
     CALL TEXTC(1,ILA23,24,19,1,1,3,IER)
     ENCODE(96,3013,ILA24)
3013 FORMAT('GOOT(LIGHT)')
     CALL TEXTC(1,ILA24,24,20,1,1,3,IER)
     ENCODE(96,3,ILA25)
   3 FORMAT('WATERLINE')
     CALL TEXTC(1,ILA25,24,24,1,1,3,IER)
     ENCODE(96,3020,ILA26)
3020 FORMAT('     EMERGENCY PERFORMANCE INDEX')
     CALL TEXTC(1,ILA26,24,14,58,1,3,IER)
     ENCODE(96,3021,ILA27)
3021 FORMAT('       TRACK       CONTROL      OVERALL')
     CALL TEXTC(1,ILA27,24,16,55,1,3,IER)
     ENCODE(96,3022,ILA28)
3022 FORMAT('       DEVIATION    EFFORT     PERFORMANCE')
     CALL TEXTC(1,ILA28,24,17,53,1,3,IER)
3023 FORMAT(F12.1,F13.1,F12.1)
 531 FORMAT(F8.1,'   ',F8.1,'   ',F8.1,'   ',F8.1,'         ',F4.2,'
    C ',F4.2,'   ',F4.2,'   ',F5.2)
     CALL VCC(1,POTK,EFLG)
     CALL WRITECLOCK(0)
     CALL STARTCLOCK
     CALL ACK(3,A41)
     CALL COMPUTE
 101 CONTINUE
 149 ICC=ICC+1
     IF(ICC.NE.1)GO TO 621
     ENCODE(96,3023,ILAP)RPERF1,RPERF2,RPERF3
```

```
      CALL TEXTC(1,ILAP,24,18,53,1,3,IER)
  621 CONTINUE
      IF(SENSE SWITCH 6)9999,9998
 9999 Z=ZZZZ
      ZI=Z
      GO TO 792
 9998 CONTINUE
      APCTK5=ABS(FCTK(5))
      IF(APCTK5.GT..1)GO TO 2070
      Z=1.4*A41
      ZI=Z
      Z7=Z
      Z8=Z
      Z9=Z
      Z10=Z
      GO TO 744
  744 CONTINUE
      IF(Z.GT.Z2)GO TO 790
      IF(Z.LT.Z3)GO TO 791
      IF(Z.GT.Z4)GO TO 792
      IF(Z.LT.Z5)GO TO 792
      Z=0.0
      ZI=Z
      Z7=0.
      Z8=0.
      Z9=0.
      Z10=0.
      GO TO 792
  790 Z=Z2
      ZI=Z
      Z7=Z2
      Z8=Z2
      Z9=Z2
      Z10=Z2
      GO TO 792
  791 Z=Z3
      ZI=Z
      Z7=Z3
      Z8=Z3
      Z9=Z3
      Z10=Z3
      GO TO 792
 2070 CONTINUE
      Z=1.4*A41
      Z7=Z
      Z8=Z
      Z9=Z
      Z10=ZAB*FCTK(5)
```

```
      IF(Z.GT.Z2)GO TO 2075
      IF(Z.LT.Z3)GO TO 2076
      IF(Z.GT.Z4)GO TO 792
      IF(Z.LT.Z5)GO TO 792
      Z=0.
      Z7=0.
      Z8=0.
      Z9=0.
      Z10=ZAE*FCTK(5)
      GO TO 792
 2075 Z=Z2
      Z7=Z2
      Z8=Z2
      Z9=Z2
      Z10=ZAE*FCTK(5)
      GO TO 792
 2076 Z=Z3
      Z7=Z3
      Z8=Z3
      Z9=Z3
      Z10=ZAE*FCTK(5)
      GO TO 792
  792 CONTINUE
      IF(ICC.NE.2)GO TO 622
      CALL VCC(1,FCTK,EFLG)
  622 CONTINUE
      T7=T7MAX*(1.+POTK(1))/2.
      T8=T8MAX*(1.+POTK(3))/2.
      T9=T9MAX*(1.+POTK(4))/2.
      T10=T10MAX*(1.+POTK(6))/2.
      T=T7+T8+T9+T10
      IF(T.GT.TMAX)GO TO 793
      IF(T.LT.0.0)GO TO 794
      GO TO 795
  793 T=TMAX
      GO TO 795
  794 T=0.0
      GO TO 795
  795 CONTINUE
      X1=PSI/6.283185
      IX=X1
      AIX=X1-IX
      IF(AIX.LT.0.0)GO TO 10
      HEAD=AIX*360.
      GO TO 11
   10 AIX1=1.+AIX
      HEAD=AIX1*360.
   11 CONTINUE
```

```
      ARATE=(360.*R)/6.283185
      ARCLL=(360.*PHI)/6.283185
      APITCH=(360.*THETA)/6.283185
      SPEED=U*.597213
      IF(SENSE SWITCH 2)175,176
  175 CONTINUE
      CALL ADK(3,A41)
      ZI=Z-SLIP*BETA
      U=U+DELT*UDOT
      V=V+DELT*VDOT
      R=R+DELT*RDOT
      P=P+DELT*PDOT
      Q=Q+DELT*QDOT
      PHI=PHI+DELT*P
      THETA=THETA+DELT*Q
      UDOT=((T/AM)*COS(ZI))-(CDX/AM)*U*U+V*R
      VDOT=((T/AM)*SIN(ZI))-((CDY/AM)*V*ABS(V))-U*R
      RDOT=((-1.0)*(T/RZ)*CC*SIN(ZI))+((CDY/RZ)*V*WW*ABS(V))
     C+(A22*U*V*WW)/RZ
      RDP=6.-50.*((SIN(PHI))/(COS(PHI)))
      RDS=6.+50.*((SIN(PHI))/(COS(PHI)))
      FSP=128000.*RDP
      FSS=128000.*RDS
      PDOT=(((FSP-FSS)*50.)-(T*(SIN(ZI)))+(CDY*V*(ABS(V))*30.)-
     C(CDZP*V*(ABS(V))*50.)-A33*U*P)/RX
      F1=192000.+3200000.*(SIN(THETA))/(COS(THETA))
      F2=192000.-3200000.*(SIN(THETA))/(COS(THETA))
      F3=CDP*FB*WE*(LD-L3*((SIN(THETA))/(COS(THETA))))
      QDOT=((T*(COS(ZI))*LF)+(F3*L3)+(F2*L3)-(CDX*U*U*LZ)-(F1*L3)
     C-(A34*U*Q))/RY
      HYU=A11*X*AU
      HYV=A21*X*AV
      HYR=A31*X*AR
      HYUDOT=((T/AM)*COS(ZI))-(CDX/AM)*HYU*HYU+HYV*HYR
      HYVDOT=((T/AM)*SIN(ZI))-((CDY/AM)*HYV*ABS(HYV))-HYU*HYR
      HYRDOT=((-1.0)*(T/RZ)*QO*SIN(ZI))+((CDY/RZ)*HYV*WW*ABS(HYV))
      B11=(HYUDOT/(AUDOT*100.))*HYBRID
      B21=(HYVDOT/(AVDOT*100.))*HYBRID
      B31=(HYRDOT/(ARDOT*100.))*HYBRID
      HYBETA=ATAN(HYV/HYU)
      HYPSI=HYPSI+DELT*HYR
      HYXODOT=HYU*COS(HYPSI)-HYV*SIN(HYPSI)
      HYYODOT=HYU*SIN(HYPSI)+HYV*COS(HYPSI)
      HYXO=HYXO+DELT*HYXODOT
      HYYO=HYYO+DELT*HYYODOT
      HYSESX=XST+((HYYO/AXST)*HYBRID)
      HYSESY=YST+((HYXO/AYST)*HYBRID)
      GO TO 177
```

```
176 CONTINUE
    CALL ACK(3,A41)
    Z7I=Z7-SLIP*BETA
    Z8I=Z8-SLIP*BETA
    Z9I=Z9-SLIP*BETA
    Z10I=Z10-SLIP*BETA
    T7C=T7*COS(Z7I)
    T8C=T8*COS(Z8I)
    T9C=T9*COS(Z9I)
    T10C=T10*COS(Z10I)
    T7S=T7*SIN(Z7I)
    T8S=T8*SIN(Z8I)
    T9S=T9*SIN(Z9I)
    T10S=T10*SIN(Z10I)
    TFCRW=T7C+T8C+T9C+T10C
    TSIDE=T7S+T8S+T9S+T10S
    TYAW=(T7C*S4)+(T8C*S3)-(T9C*S1)-(T10C*S2)-(T7S*OO)-(T8S*OO)-
   C(T9S*OO)-(T10S*OO)
    U=U+DELT*UDOT
    V=V+DELT*VDOT
    R=R+DELT*RDOT
    P=P+DELT*PDOT
    Q=Q+DELT*QDOT
    PHI=PHI+DELT*P
    THETA=THETA+DELT*Q
    UDOT=(((-1.0)*CDX/AM)*U*U)+(TFCRW/AM)+V*R
    VDOT=(TSIDE/AM)-((CDY/AM)*V*ABS(V))-U*R
    RDOT=((CDY/RZ)*WW*V*ABS(V))+(TYAW/RZ)+(A22*U*V*(WW/RZ))
    RDP=6.-50.*((SIN(PHI))/(COS(PHI)))
    RDS=6.+50.*((SIN(PHI))/(COS(PHI)))
    FSP=128000.*RDP
    FSS=128000.*RDS
    PDOT=(((FSP-FSS)*50.)-(TSIDE*5.)+(CDY*V*(ABS(V))*30.)-
   C(CDZP*V*(ABS(V))*50.)-A33*U*P)/RX
    F1=192000.+3200000.*(SIN(THETA))/(COS(THETA))
    F2=192000.-3200000.*(SIN(THETA))/(COS(THETA))
    F3=CDP*PB*WE*(LD-L3*((SIN(THETA))/(COS(THETA))))
    QDOT=((TFCRW*LP)+(F3*L3)+(F2*L3)-(CDX*U*U*LZ)-(F1*L3)
   C-(A34*U*Q))/RY
    WDOT=0.
    HYSESX=XST+((HYYO/AXST)*HYBRID)
    HYSESY=YST+((HYXO/AYST)*HYBRID)
180 CONTINUE
177 CONTINUE
    BETA=ATAN(V/U)
    VS=((U*U)+(V*V))**.5
    PSI=PSI+DELT*R
    XODOT=U*COS(PSI)-V*SIN(PSI)
```

```
      YDOT=U*SIN(PSI)+V*COS(PSI)
      XO=XC+DELT*XDOT
      YO=YC+DELT*YDOT
      SESX=XST+(YC/AXST)
      SESY=YST+(XO/AYST)
      IF(H(1).GT.XC)GO TO 700
      H(1)=XC
  700 IF(H(2).GT.YO)GO TO 701
      H(2)=YC
  701 IF(H(3).GT.ZZ)GO TO 702
      H(3)=ZZ
  702 IF(H(4).GT.PSI)GO TO 703
      H(4)=PSI
  703 IF(H(5).GT.PHI)GO TO 704
      H(5)=PHI
  704 IF(H(6).GT.THETA)GO TO 705
      H(6)=THETA
  705 IF(H(7).GT.U)GO TO 706
      H(7)=U
  706 IF(H(8).GT.V)GO TO 707
      H(8)=V
  707 IF(H(9).GT.W)GO TO 708
      H(9)=W
  708 IF(H(10).GT.R)GO TO 709
      H(10)=R
  709 IF(H(11).GT.P)GO TO 710
      H(11)=P
  710 IF(H(12).GT.Q)GO TO 711
      H(12)=Q
  711 IF(H(13).GT.UDOT)GO TO 712
      H(13)=UDOT
  712 IF(H(14).GT.VDOT)GO TO 713
      H(14)=VDOT
  713 IF(H(15).GT.WDOT)GO TO 714
      H(15)=WDOT
  714 IF(H(16).GT.RDOT)GO TO 715
      H(16)=RDOT
  715 IF(H(17).GT.PDOT)GO TO 716
      H(17)=PDOT
  716 IF(H(18).GT.QDOT)GO TO 718
      H(18)=QDOT
  718 IF(H(30).LT.XC)GO TO 719
      H(30)=XC
  719 IF(H(31).LT.YO)GO TO 720
      H(31)=YC
  720 IF(H(32).LT.ZZ)GO TO 721
      H(32)=ZZ
  721 IF(H(33).LT.PSI)GO TO 722
```

```
      H(33)=PSI
  722 IF(H(34).LT.PHI)GO TO 723
      H(34)=PHI
  723 IF(H(35).LT.THETA)GO TO 724
      H(35)=THETA
  724 IF(H(36).LT.U)GO TO 725
      H(36)=U
  725 IF(H(37).LT.V)GO TO 726
      H(37)=V
  726 IF(H(38).LT.W)GO TO 727
      H(38)=W
  727 IF(H(39).LT.R)GO TO 728
      H(39)=R
  728 IF(H(40).LT.P)GO TO 729
      H(40)=P
  729 IF(H(41).LT.Q)GO TO 730
      H(41)=Q
  730 IF(H(42).LT.UDOT)GO TO 731
      H(42)=UDOT
  731 IF(H(43).LT.VDOT)GO TO 732
      H(43)=VDOT
  732 IF(H(44).LT.WDOT)GO TO 733
      H(44)=WDOT
  733 IF(H(45).LT.RDOT)GO TO 734
      H(45)=RDOT
  734 IF(H(46).LT.PDOT)GO TO 735
      H(46)=PDOT
  735 IF(H(47).LT.QDOT)GO TO 736
      H(47)=QDOT
  736 CONTINUE
      IF(SENSE SWITCH 1)102,105
  102 CONTINUE
      IIX=IIX+1
      IF(IIX.LT.APRINT)GO TO 7654
      IIX=0
      WRITE(6,100)TIME,U,V,R,PHI,THETA,XO,YO
  100 FORMAT(6F12.5,2F8.1)
 7654 CONTINUE
 105  CONTINUE
      IF(ICC.NE.3)GO TO 623
  602 ENCODE(96,516,ILAP)T,Z,DELT,TIME,APITCH,HEAD,ARATE,BETA,AROLL,SPEE
     CD
      CALL TEXTC(1,ILAP,24,34,1,1,3,IER)
  623 CONTINUE
  58  CONTINUE
      CALL BOAT
      IF(PCTK(2).GT.0.)GO TO 2071
      ABSYO=ABS(YO)
```

```
      IF(APOTK5.LT..1)GO TO 61
      ABSYOD=ABS(YODOT)
      RPERF1=RPERF1+((YO*YO)*DELT/840000.)+DELT
      RPERF2=RPERF2+((Z*Z)*DELT*35.7)
      RPERF3=RPERF1+RPERF2
 60   CONTINUE
      IF((ABSYO.GT.ASTOP).OR.(ABSYOD.GT.AASTOP))GO TO 61
      IN=IN+1
      IF(IN.LT.60)GO TO 61
3068  CALL VCD(1,POTK,EFLG)
      IF(RPERF3.LT.250.)GO TO 3080
      IF(RPERF3.LT.500.)GO TO 3083
      ENCODE(96,3085,ILAP)RPERF1,RPERF2
3085  FORMAT(F12.1,F13.1,'       GOOD')
      GO TO 3088
3080  ENCODE(96,3081,ILAP)RPERF1,RPERF2
3081  FORMAT(F12.1,F13.1,'     EXCELLENT')
      GO TO 3088
3083  ENCODE(96,3084,ILAP)RPERF1,RPERF2
3084  FORMAT(F12.1,F13.1,'   ABOVE AVERAGE')
      GO TO 3088
3088  CONTINUE
      CALL TEXTO(1,ILAP,24,18,53,1,3,IER)
      ENCODE(96,516,ILAP)T,Z,DELT,TIME,APITCH,HEAD,ARATE,BETA,AROLL,SPEE
     CD
      CALL TEXTO(1,ILAP,24,34,1,1,3,IER)
      ENCODE(96,501,ILAP)YO,XO,ZZ,PSI,PHI,THETA
      CALL TEXTO(2,ILAP,24,10,4,1,3,IER)
      IF(POTK(2).LT.0.)GO TO 3068
      GO TO 2071
 61   CONTINUE
2071  IF(POTK(2).GT.0.0)GO TO 2080
      GO TO 2072
2080  CALL VCD(1,POTK,EFLG)
      IF(POTK(2).GT..5)GO TO 1111
      IF(POTK(2).LT..25)GO TO 23
      GO TO 1111
 23   CALL WRITECLOCK(0)
      GO TO 2071
2072  CONTINUE
      IF(ICO.NE.15)GO TO 624
      IIFR(1)=IFR(1)-1
      J(1)=J(1)+1
      IJ(1)=4+J(1)
      IJA(1)=IJ(1)+1
      IVIEWAA(1)=IHEAD(0,10)
      IVIEWAA(2)=IPACK(.0,.0,0)
      IVIEWAA(3)=IPACK(XREPET(1),YREPET(1),0)
```

```
      IVIEWAA(4)=IPACK(XREPET(1),YREPET(1),1)
      IVIEWAA(IJ(1))=IPACK(SESX,SESY,1)
      DO 2015 I=IJA(1),IPR(1)
 2015 IVIEWAA(I)=0
      IF(IJA(1).LT.IIPR(1))GO TO 2017
 2018 CONTINUE
      DO 2016 I=1,IPR(1)
 2016 IVIEWAA(I)=0
      XREPET(1)=SESX
      YREPET(1)=SESY
      J(1)=0
      IJ(1)=0
      IJA(1)=0
 2017 CONTINUE
      CALL GRAPHO(1,IVIEWAA,IPR(1),15,IER)
  624 CONTINUE
  626 CONTINUE
      IF(ICC.NE.10)GO TO 630
      IIPR(2)=IPR(2)-1
      J(2)=J(2)+1
      IJ(2)=4+J(2)
      IJA(2)=IJ(2)+1
      TIM(2)=TIM(2)+DELT
      TIMPLT(2)=(TIM(2)/SF1)-SF14
      XPLOT(2)=TIMPLT(2)
      YPLOT(2)=(U/(10.*UMAX))+.9
      IVIEWC(1)=IHEAD(0,10)
      IVIEWC(2)=IPACK(.0,.0,0)
      IVIEWC(3)=IPACK(-.9,YREPET(2),0)
      IVIEWC(4)=IPACK(-.9,YREPET(2),1)
      IVIEWC(IJ(2))=IPACK(XPLOT(2),YPLOT(2),1)
      DO 2020 I=IJA(2),IPR(2)
 2020 IVIEWC(I)=0
      IF(IJA(2).LT.IIPR(2))GO TO 2032
 2038 CONTINUE
      DO 2044 I=1,IPR(2)
 2044 IVIEWC(I)=0
      TIM(2)=0.
      YREPET(2)=YPLOT(2)
      J(2)=0
      IJ(2)=0
      IJA(2)=0
 2032 CONTINUE
      CALL GRAPHO(1,IVIEWC,IPR(2),10,IER)
  630 CONTINUE
      IF(ICC.NE.11)GO TO 631
      IIPRD=IPRD-1
      JD=JD+1
```

```
      IJD=4+JD
      IJAD=IJD+1
      DTIME=DTIME+DELT
      DTIMPLT=(DTIME/SF1)-SF14
      DXPLOT=DTIMPLT
      DYPLOT=(V/(10.*VMAX))+0.7
      IVIEWD(1)=IHEAD(0,10)
      IVIEWD(2)=IPACK(.0,.0,0)
      IVIEWD(3)=IPACK(-.9,DYREPET,0)
      IVIEWD(4)=IPACK(-.9,DYREPET,1)
      IVIEWD(IJD)=IPACK(DXPLOT,DYPLOT,1)
      DO 2021 I=IJAD,IPRD
2021  IVIEWD(I)=0
      IF(IJAD.LT.IIPRD)GO TO 2033
2039  CONTINUE
      DO 2045 I=1,IPRD
2045  IVIEWD(I)=0
      DTIME=0.
      DYREPET=DYPLOT
      JD=0
      IJD=0
      IJAD=0
2033  CONTINUE
      CALL GRAPHC(1,IVIEWD,IPRD,11,IER)
 631  CONTINUE
      IF(ICC.NE.12)GO TO 632
      IIPRE=IPRE-1
      JE=JE+1
      IJE=4+JE
      IJAE=IJE+1
      ETIME=ETIME+DELT
      ETIMPLT=(ETIME/SF1)-SF14
      EXPLOT=ETIMPLT
      EYPLOT=(R/(10.*RMAX))+0.5
      IVIEWE(1)=IHEAD(0,10)
      IVIEWE(2)=IPACK(.0,.0,0)
      IVIEWE(3)=IPACK(-.9,EYREPET,0)
      IVIEWE(4)=IPACK(-.9,EYREPET,1)
      IVIEWE(IJE)=IPACK(EXPLOT,EYPLOT,1)
      DO 2022 I=IJAE,IPRE
2022  IVIEWE(I)=0
      IF(IJAE.LT.IIPRE)GO TO 2034
2040  CONTINUE
      DO 2046 I=1,IPRE
2046  IVIEWE(I)=0
      ETIME=0.
      EYREPET=EYPLOT
      JE=0
```

```
      IJE=0
      IJAE=0
 2034 CONTINUE
      CALL GRAPHC(1,IVIEWE,IPRE,12,IER)
  632 CONTINUE
      IF(ICC.NE.6)GO TO 626
      IIPRF=IPRF-1
      JF=JF+1
      IJF=4+JF
      IJAF=IJF+1
      FTIME=FTIME+DELT
      FTIMPLT=(FTIME/SF1)-SF14
      FXPLOT=FTIMPLT
      FYPLOT=(VDCT/(10.*VDCTM))+0.7
      IVIEWF(1)=IHEAD(0,3)
      IVIEWF(2)=IPACK(.0,.0,0)
      IVIEWF(3)=IPACK(-.9,FYREFET,0)
      IVIEWF(4)=IPACK(-.9,FYREFET,1)
      IVIEWF(IJF)=IPACK(FXPLOT,FYPLOT,1)
      DO 2023 I=IJAF,IPRF
 2023 IVIEWF(I)=0
      IF(IJAF.LT.IIPRF)GO TO 2035
 2041 CONTINUE
      DO 2047 I=1,IPRF
 2047 IVIEWF(I)=0
      FTIME=0.
      FYREFET=FYPLOT
      JF=0
      IJF=0
      IJAF=0
 2035 CONTINUE
      CALL GRAPHC(1,IVIEWF,IPRF,6,IER)
  626 CONTINUE
      IF(ICC.NE.7)GO TO 627
      IIPRG=IPRG-1
      JG=JG+1
      IJG=4+JG
      IJAG=IJG+1
      GTIME=GTIME+DELT
      GTIMPLT=(GTIME/SF1)-SF14
      GXPLOT=GTIMPLT
      GYPLOT=(RDCT/(10.*RDCTM))+0.5
      IVIEWG(1)=IHEAD(0,3)
      IVIEWG(2)=IPACK(.0,.0,0)
      IVIEWG(3)=IPACK(-.9,GYREFET,0)
      IVIEWG(4)=IPACK(-.9,GYREFET,1)
      IVIEWG(IJG)=IPACK(GXPLOT,GYPLOT,1)
      DO 2024 I=IJAG,IPRG
```

```
2024 IVIEWG(I)=0
     IF(IJAG.LT.IIPRG)GO TO 2036
2042 CONTINUE
     DO 2048 I=1,IPRG
2048 IVIEWG(I)=0
     GTIME=0.
     GYREPET=GYPLOT
     JG=0
     IJG=0
     IJAG=0
2036 CONTINUE
     CALL GRAPHC(1,IVIEWG,IPRG,7,IER)
 627 CONTINUE
     IF(ICC.NE.5)GO TO 625
     IIPRH=IPRH-1
     JH=JH+1
     IJH=4+JH
     IJAH=IJH+1
     HTIME=HTIME+DELT
     HTIMPLT=(HTIME/SF1)-SF14
     HXPLOT=HTIMPLT
     HYPLOT=(LCOT/(10.*LCCTM))+0.9
     IVIEWH(1)=IHEAD(0,3)
     IVIEWH(2)=IPACK(.0,.0,0)
     IVIEWH(3)=IPACK(-.9,HYREPET,0)
     IVIEWH(4)=IPACK(-.9,HYREPET,1)
     IVIEWH(IJH)=IPACK(HXPLOT,HYPLOT,1)
     DO 2025 I=IJAH,IPRH
2025 IVIEWH(I)=0
     IF(IJAH.LT.IIPRH)GO TO 2037
2043 CONTINUE
     DO 2049 I=1,IPRH
2049 IVIEWH(I)=0
     HTIME=0.
     HYREPET=HYPLOT
     JH=0
     IJH=0
     IJAH=0
2037 CONTINUE
     CALL GRAPHC(1,IVIEWH,IPRH,5,IER)
 625 CONTINUE
     IF(ICC.NE.8)GO TO 628
     IIPRC=IPRC-1
     JO=JC+1
     IJC=4+JC
     IJAC=IJC+1
     CTIME=CTIME+DELT
     CTIMPLT=(CTIME/SF1)-SF14
```

```
      CXPLOT=CTIMPLT
      CYPLOT=(FCCT/(10.*FCCTM))+.3
      IVIEWO(1)=IHEAD(0,3)
      IVIEWO(2)=IPACK(.0,.0,0)
      IVIEWO(3)=IPACK(-.9,CYREFET,0)
      IVIEWO(4)=IPACK(-.9,CYREFET,1)
      IVIEWO(IJO)=IPACK(CXPLOT,CYPLOT,1)
      DO 2060 I=IJAO,IPRO
 2060 IVIEWO(I)=0
      IF(IJAO.LT.IIPRO)GO TO 2065
      DO 2090 I=1,IPRO
 2090 IVIEWO(I)=0
      CTIME=0.
      CYREFET=CYPLOT
      JO=0
      IJO=0
      IJAO=0
 2065 CONTINUE
      CALL GRAPHC(1,IVIEWO,IPRO,8,IER)
  628 CONTINUE
      IF(ICC.NE.9)GO TO 629
      IIPRP=IPRP-1
      JP=JP+1
      IJP=4+JP
      IJAP=IJP+1
      PTIME=PTIME+DELT
      PTIMPLT=(PTIME/SF1)-SF14
      PXPLOT=PTIMPLT
      PYPLOT=(QCOT/(10.*QCCTM))+.1
      IVIEWP(1)=IHEAD(0,3)
      IVIEWP(2)=IPACK(.0,.0,0)
      IVIEWP(3)=IPACK(-.9,PYREFET,0)
      IVIEWP(4)=IPACK(-.9,PYREFET,1)
      IVIEWP(IJP)=IPACK(PXPLOT,PYPLOT,1)
      DO 2061 I=IJAP,IPRP
 2061 IVIEWP(I)=0
      IF(IJAP.LT.IIPRP)GO TO 2066
      DO 2091 I=1,IPRP
 2091 IVIEWP(I)=0
      PTIME=0.
      PYREFET=PYPLOT
      JP=0
      IJP=0
      IJAP=0
 2066 CONTINUE
      CALL GRAPHC(1,IVIEWP,IPRP,9,IER)
      IF(IER.NE.0)OUTPUT(101) IER, '   10
  629 CONTINUE
```

```
      IF(ICC.NE.13)GO TO 633
      IIPRQ=IPRQ-1
      JQ=JQ+1
      IJQ=4+JQ
      IJAQ=IJQ+1
      QTIME=QTIME+DELT
      QTIMPLT=(QTIME/SF1)-SF14
      QXPLOT=QTIMPLT
      QYPLOT=(F/(10.*PMAX))+.3
      IVIEWQ(1)=IHEAD(0,10)
      IVIEWQ(2)=IPACK(.0,.0,0)
      IVIEWQ(3)=IPACK(-.9,QYREFPET,0)
      IVIEWQ(4)=IPACK(-.9,QYREFPET,1)
      IVIEWQ(IJQ)=IPACK(QXPLOT,QYPLOT,1)
      DO 2062 I=IJAQ,IPRQ
 2062 IVIEWQ(I)=0
      IF(IJAQ.LT.IIPRQ)GO TO 2067
      DO 2092 I=1,IPRQ
 2092 IVIEWQ(I)=0
      QTIME=0.
      QYREFPET=QYPLOT
      JQ=0
      IJQ=0
      IJAQ=0
 2067 CONTINUE
      CALL GRAPHC(1,IVIEWQ,IPRQ,13,IER)
      IF(IER.NE.0)OUTPUT(101) IER, '   9-
  633 CONTINUE
      IF(ICC.NE.14)GO TO 634
      IIPRR=IPRR-1
      JR=JR+1
      IJR=4+JR
      IJAR=IJR+1
      RTIME=RTIME+DELT
      RTIMPLT=(RTIME/SF1)-SF14
      RXPLOT=RTIMPLT
      RYPLOT=(G/(10.*QMAX))+.1
      IVIEWR(1)=IHEAD(0,10)
      IVIEWR(2)=IPACK(.0,.0,0)
      IVIEWR(3)=IPACK(-.9,RYREFPET,0)
      IVIEWR(4)=IPACK(-.9,RYREFPET,1)
      IVIEWR(IJR)=IPACK(RXPLOT,RYPLOT,1)
      DO 2063 I=IJAR,IPRR
 2063 IVIEWR(I)=0
      IF(IJAR.LT.IIPRR)GO TO 2068
      DO 2073 I=1,IPRR
 2073 IVIEWR(I)=0
      RTIME=0.
```

```
      RYREFET=RYPLOT
      JR=0
      IJR=0
      IJAR=0
 2068 CONTINUE
      CALL GRAPHO(1,IVIEWR,IPRR,14,IER)
      IF(IER.NE.0)OUTPUT(101) IER, '   8        '
  634 CONTINUE
      IF(ICC.NE.4)GO TO 635
      IIPRS=IPRS-1
      JS=JS+1
      IJS=4+JS
      IJAS=IJS+1
      STIME=STIME+DELT
      STIMPLT=(STIME/SF1)-SF14
      SXPLOT=HYSESX
      SYPLOT=HYSESY
      IVIEWS(1)=IHEAD(0,1)
      IVIEWS(2)=IPACK(.0,.0,0)
      IVIEWS(3)=IPACK(SXREFET,SYREPET,0)
      IVIEWS(4)=IPACK(SXREFET,SYREPET,1)
      IVIEWS(IJS)=IPACK(SXPLOT,SYPLOT,1)
      DO 2064 I=IJAS,IPRS
 2064 IVIEWS(I)=0
      IF(IJAS.LT.IIPRS)GO TO 2069
      DO 2074 I=1,IPRS
 2074 IVIEWS(I)=0
      STIME=0.
      SXREFET=HYSESX
      SYREFET=HYSESY
      JS=0
      IJS=0
      IJAS=0
 2069 CONTINUE
      CALL GRAPHO(1,IVIEWS,IPRS,4,IER)
      IF(IER.NE.0)OUTPUT(101) IER, '   5        '
  635 CONTINUE
      IF(ICC.NE.16)GO TO 636
  653 ENCODE(96,501,ILAP)YO,XO,ZZ,PSI,PHI,THETA
  600 CALL TEXTO(2,ILAP,24,10,4,1,3,IER)
  636 CONTINUE
      IF(ICC.NE.17)GO TO 637
  110 ENCODE(96,526,ILAP)H(31),H(30),H(32),H(33),H(34),H(35)
      CALL TEXTO(2,ILAP,24,13,4,1,3,IER)
  637 CONTINUE
      IF(ICC.NE.18)GO TO 638
  108 ENCODE(96,525,ILAP)H(2),H(1),H(3),H(4),H(5),H(6)
      CALL TEXTO(2,ILAP,24,12,4,1,3,IER)
```

```
638 CONTINUE
    IF(ICC.NE.19)GO TO 639
601 ENCODE(96,507,ILAP)U,V,W,R,P,Q
    CALL TEXTC(2,ILAP,24,18,4,1,3,IER)
639 CONTINUE
    IF(ICC.NE.20)GO TO 640
115 ENCODE(96,528,ILAP)H(36),H(37),H(38),H(39),H(40),H(41)
    CALL TEXTC(2,ILAP,24,21,4,1,3,IER)
640 CONTINUE
    IF(ICC.NE.21)GO TO 641
113 ENCODE(96,527,ILAP)H(7),H(8),H(9),H(10),H(11),H(12)
    CALL TEXTC(2,ILAP,24,20,4,1,3,IER)
641 CONTINUE
    IF(ICC.NE.22)GO TO 642
604 ENCODE(96,511,ILAP)UDOT,VDOT,WDOT,RDOT,PDOT,QDOT
    CALL TEXTC(2,ILAP,24,26,4,1,3,IER)
642 CONTINUE
    IF(ICC.NE.23)GO TO 643
120 ENCODE(96,530,ILAP)H(42),H(43),H(44),H(45),H(46),H(47)
    CALL TEXTC(2,ILAP,24,29,4,1,3,IER)
643 CONTINUE
    IF(ICC.NE.24)GO TO 644
118 ENCODE(96,529,ILAP)H(13),H(14),H(15),H(16),H(17),H(18)
    CALL TEXTC(2,ILAP,24,28,4,1,3,IER)
644 CONTINUE
    IF(ICC.NE.25)GO TO 645
    ENCODE(96,531,ILAP)T7,T8,T9,T10,Z7,Z8,Z9,Z10
    CALL TEXTC(1,ILAP,24,37,1,1,3,IER)
645 CONTINUE
    CALL READCLOCK(N)
    CALL WRITECLOCK(0)
    IF(SENSE SWITCH 5)740,741
740 DELT=DELTA
    GO TO 742
741 CONTINUE
    DELT=N/TINT
742 CONTINUE
    TIME=TIME+DELT
610 IF(ICO.LE.24)GO TO 101
    ICC=0
    GO TO 101
    END
    SUBROUTINE BOAT
    DIMENSION IVIEWA(75),FCTK(24)
    COMMON F,THETA,PHI,U,UMAX,XGX,XDX,UMUL,YO,HEAD,YHIGH,APCTK5,
   CDELT,Z,SPDLIN,AICCA,ICCA,POTK
    ABSYC=ABS(YC)
    FOR3=X1XE-.13
```

```
      FOR5=X1XA+.13
      YAWPX=2.5*R
      YAWPY=THETA
      X1X=.45*COS(PHI)
      Y1Y=.45*SIN(PHI)
      X1XA=(2.5*R)+X1X
      X11=2.5*R
      Y1YA=THETA-Y1Y
      X1XB=(2.5*R)-X1X
      Y1YB=THETA+Y1Y
      X1XC=(2.5*R)+(0.1*SIN(PHI))
      Y1YC=THETA+(0.1*COS(PHI))
      ICCA=ICCA+1.+(AICOA*(U/UMAX))
      Y1YD=(ICCA/514.)*(-1.0)+(THETA)-0.14
      Y1YE=(ICCA/514.)*(-1.0)+(THETA)-0.14
      X1XD=(X1XE-0.13)-((ICCA/514.)*0.2)
      X1XDL=(X1XB-0.13)-((ICOA/514.)*0.2)-(((ICCA/514.)*.2)*XGX*R)
      X1XE=X1XE-0.13
      X1XEL=X1XB-0.13-(((ICCA/514.)*.2)*XGX*R)
      X1XF=X1XA+0.13
      X1XFR=X1XA+0.13-(((ICCA/514.)*.2)*XDX*R)
      X1XG=(X1XA+0.13)+((ICCA/514.)*0.2)
      X1XGR=(X1XA+0.13)+((ICOA/514.)*0.2)-(((ICOA/514.)*.2)*XDX*R)
      Y1YR=Y1YA-0.14
      Y1YL=Y1YB-0.14
      Y1YEE=Y1YE-(((ICOA/514.)*.2)*((U/UMAX)*(UMUL)))
      REARA1=-0.5
      REARA2=-0.2+Y1Y
      REARB1=-0.5
      REARB2=-0.54+Y1Y
      REARC1=0.5
      REARC2=-0.2-Y1Y
      REARD1=0.5
      REARD2=-0.54-Y1Y
      SPDLIN=((-.24)*(U/UMAX))-.14
      YOREF1=(YC/10000.)*(-1.)
      YOREF=(YO/1000.)*(-1.)
      ROAD4X=ROCHED4+(1.5*SIN(YOREF))
      IF(ROAD4X.GT.1.)GO TO 30
      IF(ROAD4X.LT.-1.)GO TO 31
      GO TO 32
   30 CONTINUE
      ROAD4X=1.
      GO TO 32
   31 CONTINUE
      ROAD4X=-1.
      GO TO 32
   32 CONTINUE
```

```
      ROAD1X=-.3+RODHED4+(1.5*SIN(YOREF))
      IF(ROAD1X.GT.1.)GO TO 33
      IF(ROAD1X.LT.-1.)GO TO 34
      GO TO 35
   33 CONTINUE
      ROAD1X=1.
      GO TO 35
   34 CONTINUE
      ROAD1X=-1.
      GO TO 35
   35 CONTINUE
      ROAD2X=+.3+RODHED4+(1.5*SIN(YOREF))
      IF(ROAD2X.GT.1.)GO TO 36
      IF(ROAD2X.LT.-1.)GO TO 37
      GO TO 38
   36 CONTINUE
      ROAD2X=1.
      GO TO 38
   37 ROAD2X=-1.
      GO TO 38
   38 CONTINUE
      RODHED1=180.
      RODHED2=HEAD+180.
      IF(RODHED2.LT.360.)GO TO 42
      RODHED2=RODHED2-360.
42    CONTINUE
      RODDIF=((RODHED1-RODHED2)*3.14)/180.
      RODHED4=SIN(RODDIF)
      ROAD3X=RODHED4+ROAD4X
      IF(ROAD3X.GT.1.)GO TO 39
      IF(ROAD3X.LT.-1.)GO TO 40
      GO TO 41
   39 CONTINUE
      ROAD3X=1.
      GO TO 41
   40 CONTINUE
      ROAD3X=-1.
      GO TO 41
   41 CONTINUE
      YROAD=ABS(ROAD4X)
      ROAD3Y=YHIGH-((YROAD)*YHIGH)
      ROAD1Y=0.01
      ROAD2Y=0.01
      ROAD4Y=0.01
      IF(ABSYO.GT.1500.)GO TO 47
      IF(RODHED2.LT.270.)GO TO 46
      GO TO 47
46    IF(RODHED2.GT.090.)GO TO 45
```

```
      GO TO 47
47    CONTINUE
      ROAD1X=-1.
      ROAD2X=-1.
      ROAD3X=-1.
      ROAD4X=-1.
      ROAD1Y=0.
      ROAD2Y=0.
      ROAD3Y=0.
      ROAD4Y=0.
 45   CONTINUE
65    CONTINUE
182   CONTINUE
      IF(Y1YE.GT.SFDLIN)GO TO 79
      ICOA=0
79    CONTINUE
59    CONTINUE
      IVIEWA(1)=IHEAD(0,10)
      IVIEWA(2)=IPACK(-.5,REARA2,0)
      IVIEWA(3)=IPACK(X1XE,Y1YE,1)
      IVIEWA(4)=IPACK(X1XA,Y1YA,1)
      IVIEWA(5)=IPACK(+.5,REARC2,1)
      IVIEWA(6)=IPACK(-.5,REARA2,1)
      IVIEWA(7)=IPACK(-.5,REARB2,1)
      IVIEWA(8)=IPACK(+.5,REARD2,1)
      IVIEWA(9)=IPACK(+.5,REARC2,1)
      IF(R.GT..02)GO TO 50
      IF(R.LT.-.02)GO TO 51
      IVIEWA(10)=IPACK(X1XD,Y1YEE,0)
      IVIEWA(11)=IPACK(X1XE,Y1YE,1)
      IVIEWA(12)=IPACK(X1XF,Y1YE,0)
      IVIEWA(13)=IPACK(X1XG,Y1YEE,1)
      IVIEWA(14)=IPACK(-1.15,-.14,0)
      IVIEWA(15)=IPACK(HOR3,-.14,1)
      IVIEWA(16)=IPACK(HOR5,-.14,0)
      IVIEWA(17)=IPACK(+1.15,-.14,1)
      IVIEWA(18)=IPACK(0.,0.,0)
      GO TO 55
50    IVIEWA(10)=IPACK(X1XA,Y1YA,0)
      IVIEWA(11)=IPACK(X1XA,Y1YR,1)
      IVIEWA(12)=IPACK(.5,REARD2,1)
      IVIEWA(13)=IPACK(X1XFR,Y1YE,0)
      IVIEWA(14)=IPACK(X1XGR,Y1YEE,1)
      IVIEWA(15)=IPACK(-1.15,-.14,0)
      IVIEWA(16)=IPACK(-.5,-.14,1)
      IVIEWA(17)=IPACK(HOR5,-.14,0)
      IVIEWA(18)=IPACK(1.15,-.14,1)
      GO TO 55
```

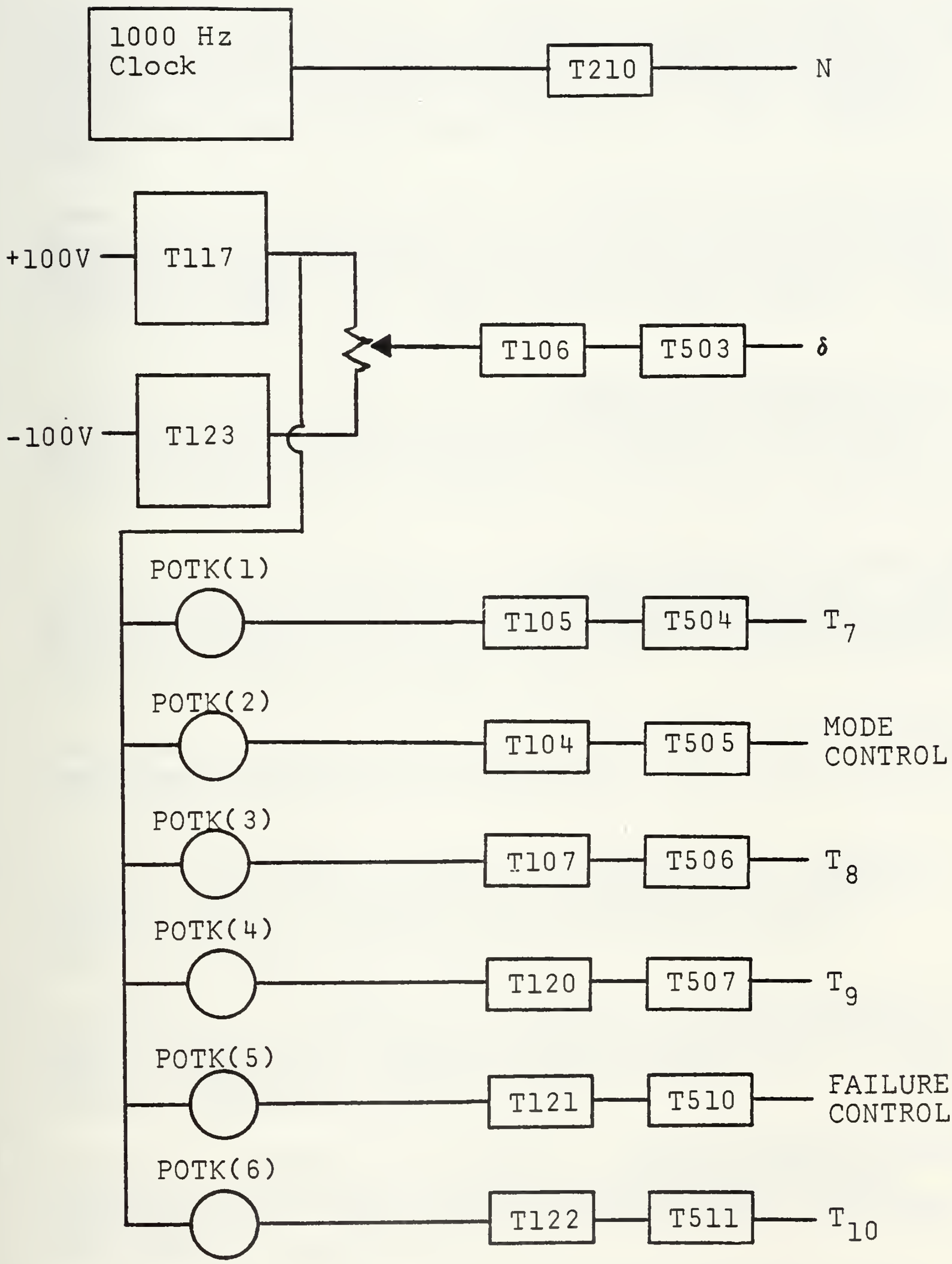
```
51    IVIEWA(10)=IPACK(X1XE,Y1YB,0)
      IVIEWA(11)=IPACK(X1XB,Y1YL,1)
      IVIEWA(12)=IPACK(-.5,REARB2,1)
      IVIEWA(13)=IPACK(X1XEL,Y1YE,0)
      IVIEWA(14)=IPACK(X1XCL,Y1YEE,1)
      IVIEWA(15)=IPACK(-1.15,-.14,0)
      IVIEWA(16)=IPACK(HCR3,-.14,1)
      IVIEWA(17)=IPACK(.5,-.14,0)
      IVIEWA(18)=IPACK(1.15,-.14,1)
55    CONTINUE
      IVIEWA(19)=IPACK(YAWFX,YAWPY,0)
      IVIEWA(20)=IPACK(YAWFX,YAWPY,1)
      IVIEWA(21)=IPACK(YAWFX,YAWPY,1)
      IVIEWA(22)=IPACK(YAWFX,YAWPY,1)
      IVIEWA(23)=IPACK(ROAD1X,ROAD1Y,0)
      IVIEWA(24)=IPACK(ROAD4X,ROAD4Y,1)
      IVIEWA(25)=IPACK(ROAD2X,ROAD2Y,1)
      IVIEWA(26)=IPACK(ROAD3X,ROAD3Y,1)
      IVIEWA(27)=IPACK(ROAD3X,ROAD3Y,1)
      IVIEWA(28)=IPACK(ROAD1X,ROAD1Y,1)
      IVIEWA(29)=IPACK(ROAD4X,ROAD4Y,0)
      IVIEWA(30)=IPACK(ROAD3X,ROAD3Y,1)
      IVIEWA(31)=0
      CALL GRAPHC(1,IVIEWA,31,3,IER)
3333  CONTINUE
      RETURN
      END
$
¬META9300 SI,LC
X1       EQU      1
X2       EQU      2
A        EQU      5
B        EQU      4
$ADDA    PZE
         BRM      $SETUPN
         PZE      4
AD       PZE
NAD      PZE
DA       PZE
NDA      PZE
         SKR      FINFLG
         BRU      $-1
         LDA      *NDA
         MRG      #077700000
         COPY     %A,X1<
         BRX      $82,1
         BRU      ATCD
         STZ      DCH
```

```
        BRU     $+2
RTCI    MFT     DA
        MFC     DCH
        LDP     *DA
        LLSE    15
        ARSE    15
        CCPY    C777,%B,X2<,%C,B<
        CCPY    %X2,A<,%A,X2<
        CCPY    %-A,A<
        CCPY    %X2,A<,%A,X2<
        AFSD    0,2
        ETR     #C77777CCC
        ADD     DCH
        STA     DACCM,1
        BRX     RTCI,1
        STZ     FINFLG
        LDA     ERMFIN
        STA     C41
        LDA     *NDA
        LLSA    15
        ADD     #DACOM
        STA     DACW
        ECM     C35001
        PCT     DACW
ATCD    LDA     *NAD
        STA     CCUNT
        SKR     CCUNT
        BRU     $+2
        BRR     ADDA
        LLSA    15
        ADD     #ADCOM
        STA     ADCW
        SKN     FINFLG
        BRU     $-1
        STZ     FINFLG
        LDA     ERMFIN
        STA     040
        ECM     C34C01
        PCT     ADCW
        LDX     #00100000,1
        SKN     FINFLG
        BRU     $-1
ITCR    LDA     ADBUF,1
        CCPY    %C,B<
        FLA     0.0
        STD     *AD
        MFT     AD
        SKR     CCUNT
```

```
         BRX      ITCF,1
         BRR      ACCA
BRMFIN   BRM      FIN
FIN      PZE
         SKR      FINFLG
         BRU      $-1
         BRC      *FIN
FINFLG   PZE
DCH      PZE
CCUNT    PZE
CON      FCRM     9,15
ADCW     PZE
ADCCM    CCN      0,ACBUF
         DATA     C,0,0,0,0,0,C,0,0,0,0,0,0,0,0
         DATA     C,0,0,0,0,0,0,0,0,0,0,00,0,0
         CATA     C,0,00,0,0,C,0,C,0,C,0,0,0,C,0
ADBUF    RES      32
DACW     PZE
DACCM    RES      13
         END
¬EOF
$
¬ASSIGN X1=MT2A
¬LOAC XR,MAF
¬SAVE
¬DATA
```

APPENDIX C

RTS5D Wiring Diagram



LIST OF REFERENCES

1. Kaplan, P., Bentson, J., Sargent, T.P., A Study of Surface Effect Ship (SES) Craft, Loads and MOtions, Part I, Equations of Motion of SES Craft with Six Degrees of Freedom, Contract No. C-1-35055, Report No. 71-84A, Oceanics Inc., Technical Industrial Park, Plainview, N.Y., August 1971.

2. Gerba, A. Jr., Thaler, G.J., A Method for Scaling the Heave Motion Equations of the C.A.B. 6-DOF Loads and Motions Program from Model 70 Full Size Craft, Progress Report to SESPO, U.S. Naval Postgraduate School, Monterey, California, December 1977.

3. Gerba, A. Jr., Thaler, G.J., Sensitivity Analysis of the XR-3 Heave Equations in Plenum Air Leakage and Influx Rates, Progress Report to SESPO, U.S. Naval Postgraduate School, Monterey, California, September 1978.

4. Gerba, A. Jr., Thaler, G.J., Development of a Real Time Hybrid Computer Simulation for the 3K-SES, 500F, Data-Based Program, Part I - Familiarization and Planning, Progress Report to SESPO, U.S. Naval Postgraduate School, Monterey, California, July 1979.

5. Department of the Navy, Code 1630: PRV, Surface Effect Ship Data-Based Maneuvering Simulation, SES Project PMS-304, September, 1979.

6. Lockheed Missiles and Space Company, Inc., Surface Effect Ships Aero/Hydrodynamics Technology Design Manual, Vol. I, Contract No. N00024-73-A-0919, Task Order No. 0003, May 1974.

7. Greensite, A., Pascol, L., Wiemer, M., 3K SES Ship Control System Design Information Report, Volume 1, CDRL No. E05P, Document No. TER023A, Rohr Marine, Inc., Chula Vista, California, 24 July 1978.

8. Limes, R., Thomas, W., Wong, A., Naval Postgraduate School Computer Laboratory, Manual for the FORTRAN User, Section I, U.S. Naval Postgraduate School, Monterey, California, January 1979.

9. Greensite, A.L., 3KSES Ship Control System Design Information Report, Supplement B, Stability Analysis of Maneuver Control Law, CDRL No. E05P, Document No. TER023W, Rohr Marine, Inc., Chula Vista, California, 8 October 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor Alex Gerba, Jr., Code 52Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
5.	Professor George J. Thaler, Code 52Tr Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
6.	Professor D.M. Layton, Code 57ln Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
7.	Mr. H. Robertson PMS 304, Code 141 Surface Effects Ships Project Office PO Box 34401 Bethesda, Maryland 20034	6
8.	MAJ. Thomas S. Nelson, USMC 724 Highland Ave Landis, N.C. 28088	1

Thesis
N366845 Nelson
c.1
186992
Real time stimula-
tion and control 3000
ton surface effect
ship.